PROJECT 3

CREATE A PROGRAMMABLE ELECTROMECHANICAL DIY TOY KIT

PROJECT REPORT

European Project Semester

Spring 2021

Julie Mallard

Kasper Ruys

Thomas Combes

Verena Hanner

Institute for Information Technology
Post address: Postboks 4 St. Olavs plass, 0130 Oslo
Visiting address: Holbergs plass, Oslo
Phone: 22 45 32 00

PROJECT LETTER

AVAILABILITY

European Project Semester

MAIN PROJECT TITLE:	DATE
CREATE A PROGRAMMABLE ELECTROMECHANICAL DIY TOY KIT	02.06.2021
	NUMBER OF PAGES/APPENDICES 68 / 10
PROJECT TEAM MEMBERS Julie Mallard, Kasper Ruys, Thomas Combes, Verena Hanner	INTERNAL SUPERVISOR(S) Alfredo Carella

EXTERNAL SUPERVISOR	CONTACT PERSON
MakeKit	Steinar Holøs

ABSTRACT

There is a general consensus that students need to do things themselves to experience the things they are taught. The aim of the project is to develop a new modular DIY kit that enables elementary school children to develop their creativity and coding skills in a fun and exciting way.

In collaboration with the company MakeKit the group decided to develop a modular DIY kit named Battle:bit. This kit allows one to build and code one battle robot. Children can choose between three majors design but also have the possibility to create their own. Then they can organise battles with the different robots they have created and try to find the best design and the best code to win.

The report explains the different stages in the creation of the robots, the mechanical solutions the team had to find and the development of applications and codes to control the robots. The group also extended the project by starting the creation of a progressive web application that controls the robot and thinking about instructions to allow the user to easily build it at home.

This project resulted in a successful partnership between international OsloMet students from all over Europe and MakeKit, as they were able to develop a working robot with three different designs and meet the company's expectations.

Acknowledgement

The team would like to thank first of all the **University of OsloMet** and our professors **Tengel Aas Sandtrø** and **Tom Muir** for hosting us in the European Project Semester 2021. We all liked the study environment even if the pandemic situation did not facilitate having a team project. OsloMet and the Pilestredet campus are a very pleasant place to work. Thanks to the University, we were able to build a real product using the machines and tools of the Makerspace. We would also like to thank **Evin Güler**, the manager of the Makerspace, who introduced us to the place and allowed us to use all the machines.

Without MakeKit we would never have been able to do this project and they have been a great help. We would like to thank **Steinar Holøs** and **Henning Pedersen** for the time they spent with us to create the Battle:bit kit and for all their advice and feedback.

Finally, we would like to thank our supervisor **Alfredo Carella** for his involvement in the project and the time he spent helping us with every type of question or problem we had to deal with. He helped each of us to find the right place in the project.

Contents

1. INTRODUCTION 1

1.1. The Mission 1
1.2. Gameplay 2
1.3. MakeKit 4

1.3.1. MakeKit's products 5

2. METHODS 7

2.1. Files storage 7
2.2. Meetings 8

2.2.1. In the group 8
2.2.2. With our supervisor 8
2.2.3. With MakeKit 8

2.3. Project planning 9

3. CREATIVE RESEARCH 11

3.1. Specifications 11
3.2. Concepts constraints 13
3.3. Main components available 13
3.4. First concept: the elephant 17
3.5. Second concept: the droid 18
3.6. Third concept: the scorpion 19

4. DESIGN 21

4.1. Skeleton 22

4.1.1. 3D modelling software 22
4.1.2. Fixation systems 22
4.1.3. How to fix servos motors on wood? 24

4.1.4. “Micro:bit card” principle 25
4.1.5. Equipment 28
4.1.6. Evolutions 30
4.2. Cardboard body 32
4.2.1. Mission 32
4.2.2. How to fit on the wooden skeleton? 32
4.2.3. Steps of design 32

5. CODES & INSTRUCTIONS 35

5.1. Makecode 35
5.2. Android App 36
5.3. Progressive Web App 36
5.4. Instructions 37

6. RESULTS 39

6.1. The wood skeleton 40
6.2. The cardboard body 41
6.3. The Web App 44
6.4. Instructions 44

7. CONCLUSION 45

8. APPENDICES 46

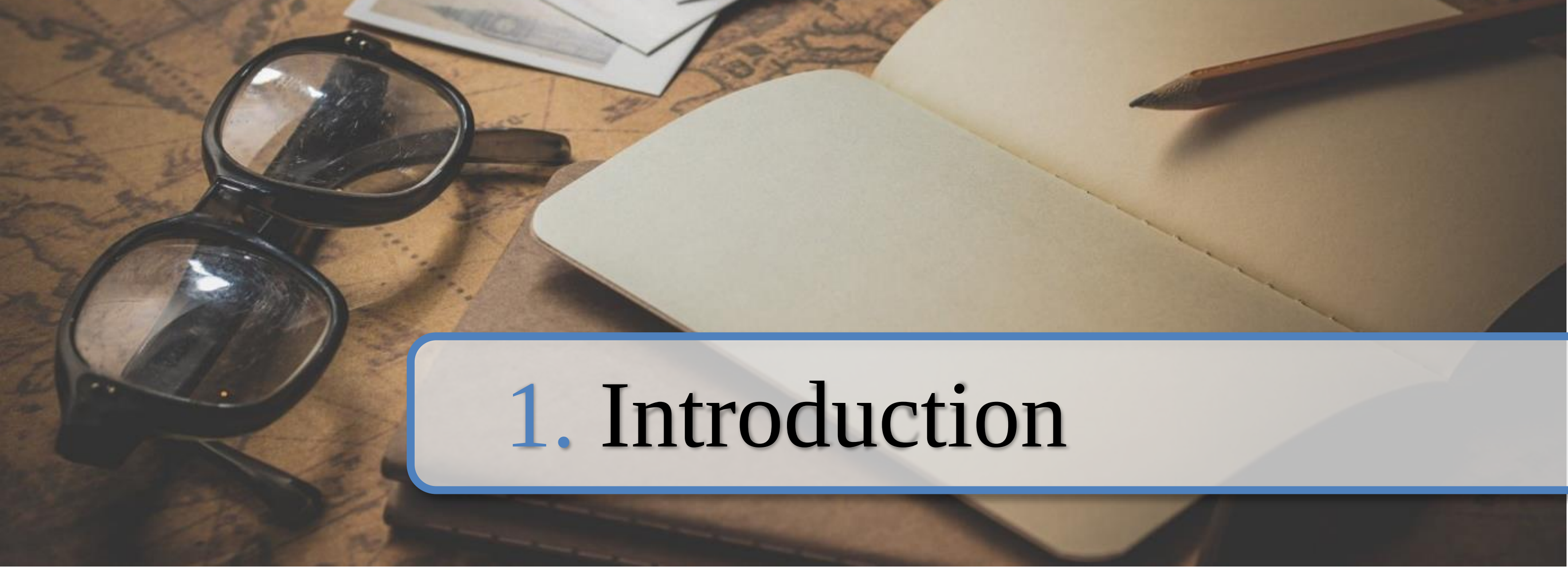
8.1. Links 46
8.2. Team 46
8.3. Project plan 48
8.4. Stakeholders analysis 49
8.5. Battle:bit project description 51

9. BIBLIOGRAPHY 57

List of Figures

Figure 1: Robot sumo battles 3
Figure 2: Second way to win - push the other out of the arena 3
Figure 3: First way to win - overthrow the other 3
Figure 4: MakeKit's kit example 5
Figure 5: Micro:bit 5
Figure 6: Wonder:kit 5
Figure 7: MakeKit's location 8
Figure 8: Planning overview 9
Figure 9: 3mm aircraft grade plywood 14
Figure 10: Solenoid 14
Figure 11: Motor 15
Figure 12: Continuous servo 15
Figure 13: Torque illustration 16
Figure 14: Micro servo 16
Figure 15: Micro:bit V2 elements description 16
Figure 16: Battery 17
Figure 17: Multi:bit control board 17
Figure 18: Concept 1 17
Figure 19: Concept 2 18
Figure 20: Cardboard holder 19
Figure 21: Concept 3 19
Figure 22: "MakeKit's system" connector secured with an O-ring 23
Figure 23: Male / Female 23
Figure 24: Small O-ring usage example 23
Figure 25: Ears specifications for small O-rings 23

Figure 26: Normal ears specifications 23
Figure 27: Ears specification for wider objects 23
Figure 28: Nylon nuts 24
Figure 29: Nylon screw 24
Figure 30: Spacer 24
Figure 31: Servo fixation 1st solution 24
Figure 32: Servo fixation 2nd solution 24
Figure 33: Continuous servos and micro servos measurement 25
Figure 34: "Micro:bit card" version 1 26
Figure 35: Block assembly 27
Figure 36: "Micro:bit card" version 2 27
Figure 37: "Micro:bit card" final version 27
Figure 38: Equipment 2 28
Figure 39: Equipment 1 28
Figure 40: Equipment 5 29
Figure 41: Equipment 4 29
Figure 42: Equipment 3 29
Figure 43: Large connector evolution 31
Figure 44: Servo holder evolution (version 2 is missing) 31
Figure 45: Base frame evolution 31
Figure 46: Third leg evolution 31
Figure 47: Animal's shape ideas 33
Figure 48: Battle:bit kits possibilities 39
Figure 49: Wooden parts inventory 40
Figure 50: Cardboard base for concept 2 41
Figure 51: Cardboard base for concept 3 41
Figure 52: Cardboard base for concept 1 41
Figure 53: Scorpion tail 42
Figure 54: Elephant face 42
Figure 55: Rhinoceros face 42
Figure 56: Robot face 43
Figure 57: Cross design 43
Figure 58: Lines design 43



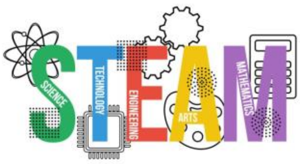
1. Introduction

“Create a programmable electromechanical DIY toy kit” is a joint project between OsloMet and MakeKit, a private company. MakeKit makes DIY kits for creative exploration that you build and code, creating learning experiences for children at home and in school. Their products give access to a new learning process where children learn by playing with diverse educational materials. Moreover, the building process of the kit creates a multidisciplinary learning experience that can be difficult to produce with standard teaching methods. Our goal with this project is to create a new kit that can fit in the MakeKit ecosystem. In addition, we want to add new competences that are not present in MakeKit’s product. Our multidisciplinary team is perfect to explore these new aspects with the various skills we have.

1.1. The Mission

The aim of the project is to develop a new kit of programmable electromechanical hobby toys. The kit is designed to give knowledge in multiple branches at home or in schools for young people. MakeKit includes the STE(A)M subjects in all their products and they are included in our project too. STEM is a curriculum based on the idea of educating student in four disciplines:

- Science
- Technology
- Engineering
- Mathematics



Compared to the STEM education, STE(A)M education integrates the Arts such as music or visual art. Instead of teaching these disciplines as separate and discrete subjects, STE(A)M integrates them into a cohesive learning experience, based on real-world application. With this approach, students are more interested in all of these disciplines and get a better experience for a multidisciplinary future job. We need to maintain this approach for our project. All disciplines need to be present so it can keep the attention of the students. This will also promote teamwork.

We have decided to create a modular DIY battle robots kit named Battle:bit.

The idea of modularity means that with the same elements and parts, students can create different robots with different characteristics. The modular robot will allow students to think about what they prefer to build and make their own choices to create the best one. In fact, the different modular robots can compete against each other in a game where the other must be pushed out of an arena or stop the other robots by knocking them over. To succeed, students can add tools and defences to their chosen design by using cardboard elements and coding their attacks.

Knowing how to code, or at least understanding it, is becoming more and more important as technology advances. But there is still no proper method implemented in schools to teach children programming in a basic but fun way.

We want to develop a learning package that will give children a basic understanding of coding and design in order to stimulate children's interest in engineering fields.

1.2. Gameplay

We have based our project on robot sumo battles. This game was invented in Japan. It makes two robots compete against each other; they are placed in a circle where the goal is to push the other one out of the circle. For a majority of the time, these robots use wheels to move and to push opponents. With the exception being tracks. The rules are simple: the robots are placed face to face and can move after five seconds. They are generally autonomous. The first one pushed out of the circle loses the battle. There are different categories of robots with different sizes and weights. It is also possible to have different arenas depending on the diameter. This game is very popular in Japan where they even organise tournaments.

For our project we adapted the rules to make the game more fun and dynamic. We kept the main rule, if you leave the arena you lose, but you have another way to win: if you knock the opponent's robot on its side you win as well. In our version, we preferred to have a controlled robot instead of an autonomous robot because we find it more interesting, engaging and less complex to program. The game will become more dynamic and surprising if we can control it.

In addition, this system gives the possibility to the children to improve their skills when they play.

We wanted to keep very simple rules for this game because the majority of the time something simple is more amusing than something complex. In addition, it should be suitable for a large number of children, from 8 to 16 years old.

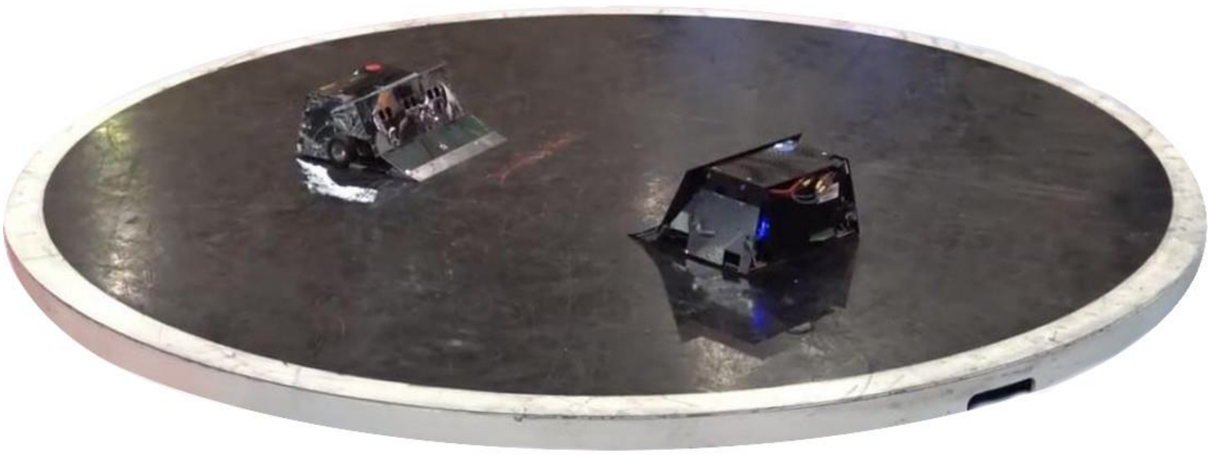


Figure 1: Robot sumo battles

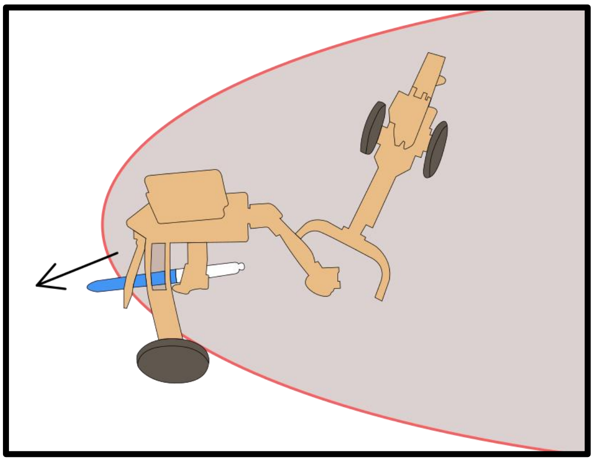


Figure 2: Second way to win - push the other out of the arena

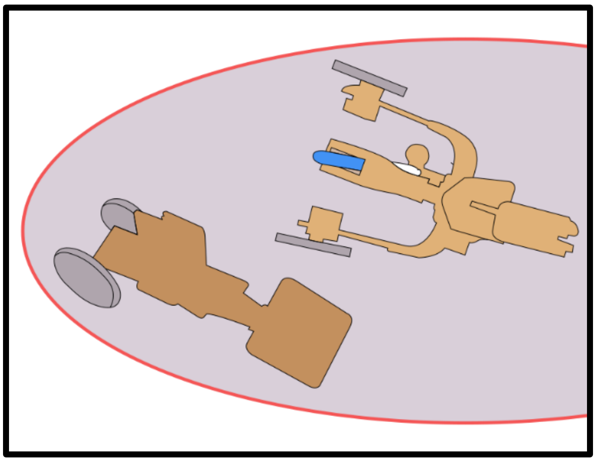


Figure 3: First way to win - overthrow the other

1.3. MakeKit

MakeKit is a privately owned company that creates DIY kits for children at home and in schools to teach young people to build, write codes and be creative in a fun and playful way. MakeKit's core idea is MAKE, PLAY, LEARN, they want to make education relevant, interesting and inspiring.

MakeKit was founded by Steinar Holøs, who is the Chief Executive Officer, and Henning Pedersen, who is the Chief Product Officer. They introduce a new way of making education relevant and interesting in technology and in the art of creating by making it accessible for children and even for parents and teachers.

They also focus on the Sustainability side by integrating four United Nations sustainable development goals:

- **Quality education:**

  MakeKit's goal is to offer a new way of learning and create new interest in sometimes neglected subjects. Education is about the joy of exploration and creativity.

- **Industry, innovation and infrastructure:**

  They try to find innovative ways to produce their products.

- **Responsible consumption and production:**

  They use environmentally friendly materials to create their products.

- **Climate action:**

  They use locally produced materials and create reusable and repairable kits.

1.3.1. MakeKit’s products

The first kit to be launched was the Air:bit in 2019, a drone using a micro:bit as microcontroller. All the other products use the same characteristics and explore diverse usage of the micro:bit. Each product gives a possibility to build, code, and develop your own design. Building and coding with the kits increases learners’ skills across a range of areas such as electronics and mathematics.

The kit form gives possibilities to the owner to create or modify parts to custom their product. Moreover, all products are reusable and repairable. To respect the United Nations sustainable development goal, it uses environmentally friendly materials such as nylon screws and nuts, silicone O-rings and majority of parts are made in wood with a laser cutter.

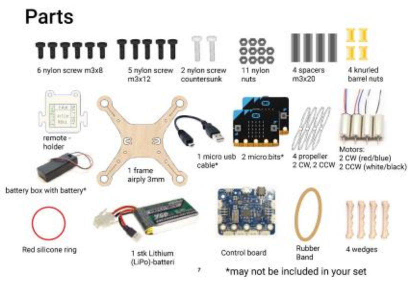


Figure 4: MakeKit's kit example

MakeKit has decided to build their products around the micro:bit, a microcomputer made for use in education. It has integrated sensors, buttons and LED-screen and can be connected to peripherals like motors, camera or sensors. It is very easy to code compared to other microcomputers because it uses a systems block coding.

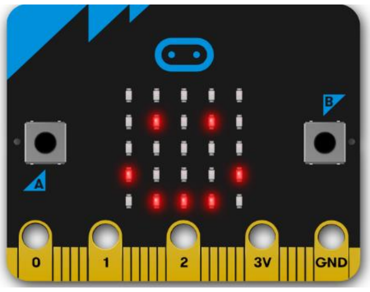


Figure 5: Micro:bit

Kits are available for both home usage and school usage but with different formats. Schools can use the Wonder:kit which includes the main parts to make 10 Explorer. To be able to build the other products, you need to buy an extension which completes the Wonder:kit with the missing parts.

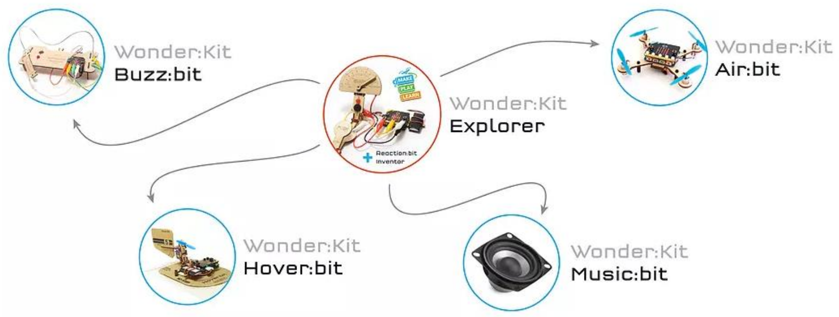
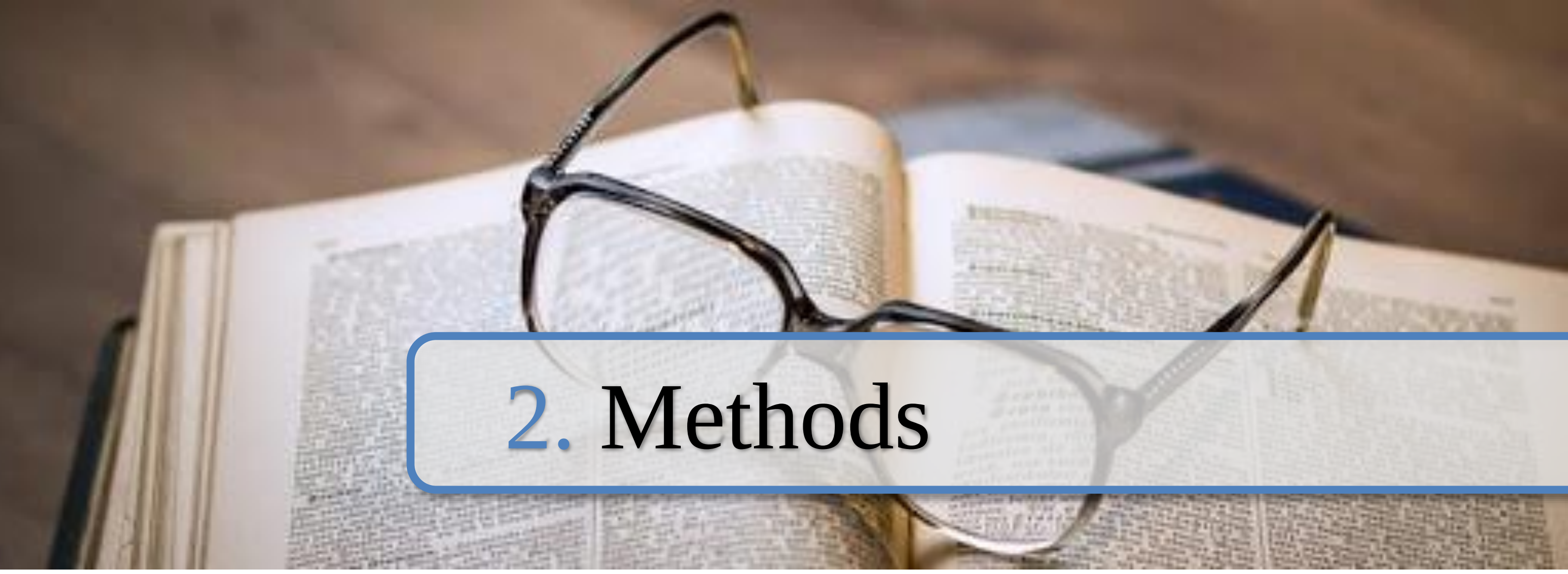


Figure 6: Wonder:kit



2. Methods

The chapter's goal is to explain how we work inside the group. We are going to talk about the project plan and the method put in place to communicate. Indeed, good communication is the key to the success of a project. With the Covid 19 crisis, when physical meetings weren't possible, we had to use other techniques to ensure this communication.

2.1. Files storage

We chose to share our files on a google drive. It allows a big storage capacity and works as a normal file explorer accessible from all devices. It is also easy to give access to an external person to specific files with different permissions.

A big challenge with shared storage is to keep track of all changes. Google Drive creates different versions when someone makes changes on one file. It is possible to know:

- Who has modified the file
- When it was modified
- What was modified

We also have access to the older version if we need to restore it.

Another advantage is the possibility to create and edit documents directly on the drive such as text, presentations or sheets. It is what we used to write reports or create slide presentations because it gave us all the possibilities to work on the same document at the same time.

2.2. Meetings

2.2.1. In the group

We made as many physical meetings as possible especially for brainstorming, the research phase and prototypes testing. The best place to meet was the Makerspace where we had the possibilities to discuss and create prototypes at the same time.
For online meetings and discussions we chose to use Discords and WhatsApp. We used WhatsApp a lot to chat and share ideas, photos and documents and sometimes used Discords for online meetings. It was important for us to meet physically at least once a week to take stock and make decisions.

2.2.2. With our supervisor

Our supervisor Alfredo Carella set up a weekly meeting at the beginning of the project. This was a Zoom meeting to discuss the progress of the projects and to find out if we needed help, for example, to access equipment or find new ways of using our skills.

2.2.3. With MakeKit

As far as meetings with our supervisors were concerned, we had an online meeting on Whereby with Steinar Holøs and Henning Pederson every Monday to show our progress and get feedback on our work. We had also created a messenger group to share our improvements and ask for advice or information. Having the possibility to chat on messenger allowed us to discuss only the important things during the meetings and to reduce the meeting time.

We have planned a few physical meetings at MakeKit to pick up parts and present and test our prototypes but we tried to limit them due to the Covid 19 pandemic. Also, MakeKit is located south of Oslo and it takes time to get there.

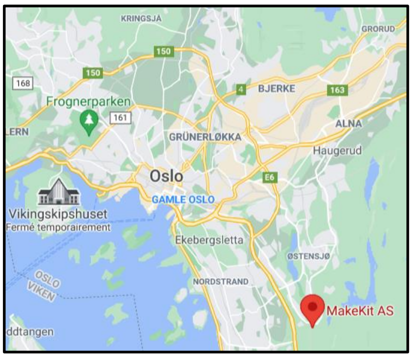


Figure 7: MakeKit's location

2.3. Project planning

To ensure the success of our project, we first built a schedule of the tasks we needed to accomplish to address MakeKit's problem. With the help of a Gantt chart, we were able to fit all the tasks into the 4-month schedule of the European Project Semester. We made the choice to realize the Gantt diagram on a google drive sheet so it can be visualized and edited by everyone. The plan is available in a larger format in appendix p 48.

Our plan is divided into seven parts that can be handle by one or several person in the group depending on their skills:

- Choosing the project : Brainstorm on the Battle:bit and its specifications (Everyone)
- Research : Research on MakeKit's ecosystem and on diverse process like laser cutting (Everyone)
- Conception : Find the three concepts and the technical solutions (Everyone)
- Design : Create the product on a 3D software to be able to cut it on laser cutter (Thomas)
- Detailing : Test and improve prototypes (Everyone)
- Production : Create the files need to produce the kit (Thomas and Julie)
- Packaging : Create diverse cardboards shapes to improve the visual (Julie)
- Documentation : Make the instruction to build and code the robot (Verena)

Kasper also works on applications to control robots with common devices like smartphones or computers.

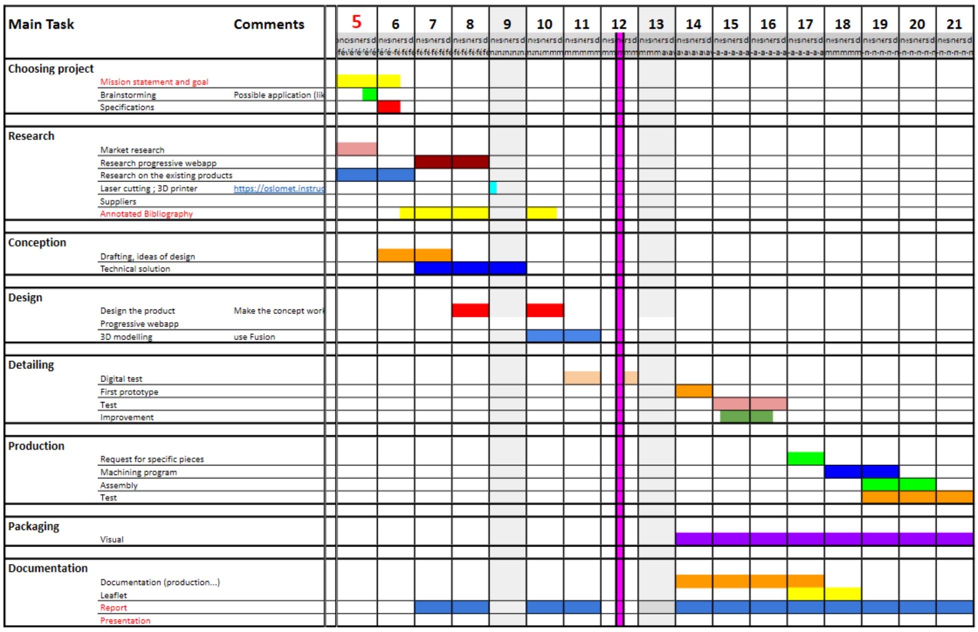


Figure 8: Planning overview

3. Creative research

At the beginning of the project, we wrote a project description with MakeKit (appendix p 51) which describes the characteristics of the product and its main functions. We can also find all the design principles used by MakeKit. We used this document as a starting point for developing the specifications and finding our three concepts. In this chapter we explain the product specifications and constraints. We also describe the concepts and the choices we made depending on the available materials.

3.1. Specifications

The goal of this project is to create a DIY robot with different designs in one kit.
The main functions are:

- Develop at least one major design, if it is possible two different, well balanced designs.
- Create functional robots for a fair battle
- Create a leaflet that can be understood by a wide audience because our target group is teachers as well as children.

General functions	Functions	Criteria	Importance
Create a DIY kit	Create functional robots	Fair battles	MH
	Create different wood designs	3 major designs	MH
	Be attractive	Shape, colour	MH

	Create a cardboard body	Easy to build	MH
Inform	Match the company's existing products	Materials, tools, performance	NTH
	Create a leaflet	User-friendly for a wide audience	MH
Protect	Resist to a fall	High of 50 cm	NTH
	Resist to a shock	Shocks from the other cars	NTH
	Resist to wear	From one year to the next	MH
Sustainability	Use recyclable materials	Wood and cardboard	NTH
	Use the packaging to create the cardboard body	Primary cardboard packaging	NTH
	Be reusable	From one year to the next	MH
Ease of use	Be transportable	No more than 500 g	MH
	Be prehensile	Size and volume of the robot	MH
Automate	Be produced by a machine	Laser cutter	MH

*MH: Must Have
**NTH: Nice To Have
***UN: UNnecessary

After making a list of the functions we want in the product, we discussed the priority of these functions. Some of these, such as creating a functional robot or being reusable are essential for the robots to function properly and to respect the wishes and image of MakeKit. We also want to have more functions such as the use of recyclable materials, but not prioritizing these functions does not harm the purpose of the project.

3.2. Concepts constraints

For the Battle:bit project, we wanted to have the possibility to build three major concepts with the same parts. These robots will be able to move forward, backward and rotate using its two wheels. Like the original sumo robot, they will be able to push their opponents simply by using the force of their movement, but in order to knock another robot to the side, each robot will be equipped with a "tool" to push, kick or grab for example. It would be nice if every piece of equipment could go on every robot or be adapted for it. The goal is to be able to build every concept with one kit instead of having one kit for each concept. With this system, it is possible to build, disassemble, and build another one very easily. You also have the possibility to adjust or upgrade your robot to give them other particularities. One idea to personalize robots was to put a cardboard body on them that can be painted or cut in different shapes such as animals or droids for example.

An essential part of the work with these three concepts is to have a fair battle between the different robots. Each robot must be able to win against another one. The way of coding a robot and controlling it needs to have the same amount of impact on the strength of a robot as the manner of its assembly. For this, each robot needs to have the same stability and the same weight. One way to secure that is using the same parts to build each design. For the stability problem, each robot needs to have a different dynamic to be really interesting, so we want to prevent them from having the same weaknesses. Our aim has been to give them different agility, stability, power and adjust their weaknesses or advantages, so they can have the same chance of winning and be totally different at the same time.

3.3. Main components available

MakeKit gave us the possibility to use several components or materials to build the robot. They have influenced us when brainstorming concepts because we tried to find a good compromise between having the best possible robots while trying to use common parts.

We used wood like MakeKit does to create our parts. They use laser cutting to cut out wooden plates that only allow for 2D parts and engraving. This technology gives us enough modularity and precision to develop and produce kits. It's a fairly quick process that produces little waste if space is optimised, but we have to be ingenious in the design phase to be able to fit the parts together. MakeKit uses mainly 3mm aircraft grade plywood with 6 layers that is a light and strong material. Its real thickness is between 3 mm and 3.6 mm and must be taken into account during the design. For prototyping we decided to use 3 mm MDF that is less expensive and weaker than plywood. It is perfect to test the solidity and be sure that the final robot is strong enough. Moreover it has the same characteristics for cutting and assembly.

Figure 9: 3mm aircraft grade plywood

We also had the possibilities to use 3D printing in case it would have been too difficult to create parts in woods because of the process limitations. MakeKit uses environmentally friendly and recyclable plastics named PLA (Polylactic acid) for 3D printings but this process is slower than laser cutting and more expensive. We tried not to use it by finding other ways to obtain the same characteristics with laser cutting.

To bring controllable movement to the robots we had four different activators to choose from, that can be used in different context depending on their characteristics:

- **Solenoid:**

It consists of a magnet surrounded by a coil. When you put a current in the coil, it creates an electromagnetic field which pushes the magnet and creates a translating movement. The return movement is generated with a spring when the current is cut.

This activator is very powerful but has a limited range. Moreover, it has limited states and it is heavy compared to other activators.

Figure 10: Solenoid

- **Motor:**

Motor creates high speed rotation with direct current. This type of motor exists in two versions, clockwise (CW) or counter clockwise (CCW). We can not change the direction of rotation without a specific control system placed before. It has the advantage to create high speed rotation (up to 46 000 rotations per minute) but with low torque. Moreover, the fixation system is not optimized for wooden parts but can easily be used with plastic parts. This kind of motor is mainly used for Air:Bit with propellers.

Figure 11: Motor

- **Servomotor:**

They are composed of one motor and one reducer to be able to transmit high torque with low speed rotation. This system is commanded with a potentiometer and requires special signal input to work compared to other systems.

It exists two types of servo motors:

➢ **Continuous servo** which is able to go forward and backward and makes full revolution. **FS90R** model is able to turn at 110 rotations per minute with no load and at 4.8 V entry voltage. It is controlled with a pulsation signal. Pulse width range is going from 900 to 1500 µs for clockwise rotation and from 1500 to 2100 µs for counter clockwise rotation. The stop signal is at 1500 µs with a dead band width of 90 µs. With a torque of 1,3 kg/cm, the best usage of this servo for our project is for turning wheels. It can be used to move our robot forward or backward. It is sold with a plastic wheel of 6 cm outer diameter and a rubber tyre. His weight is really interesting too because it is only 9g, so it is the lightest activator that we have.

Figure 12: Continuous servo

Figure 14: Micro servo

➢ In contrast to continuous servo, **micro servo** can not make full revolution rotation, but they are commanded with angle instruction. The **SG90** can go from -90° to +90° with an accuracy of 5°~10°. It is controlled with a PWM (Pulse width modulation) signal which sets the servo's angle position. It has a better torque than the continuous servo (2,5 kg/cm against 1,3 kg/cm). It means that this servo can move a load of 2,5 kg at a distance of 1cm. It has the same weight as the continuous servo FS90R.

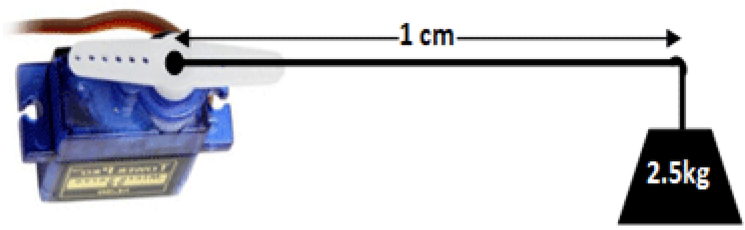


Figure 13: Torque illustration

These activators are controlled by sending signals with the micro:bit but we use the multi:bit control boards to plug in and power up the previous elements. Indeed, the micro:bit is equipped with a lot of sensors, buttons, LED lights but it can not directly power up our activators. For this we use the five pins to connect the multi:bit control board.

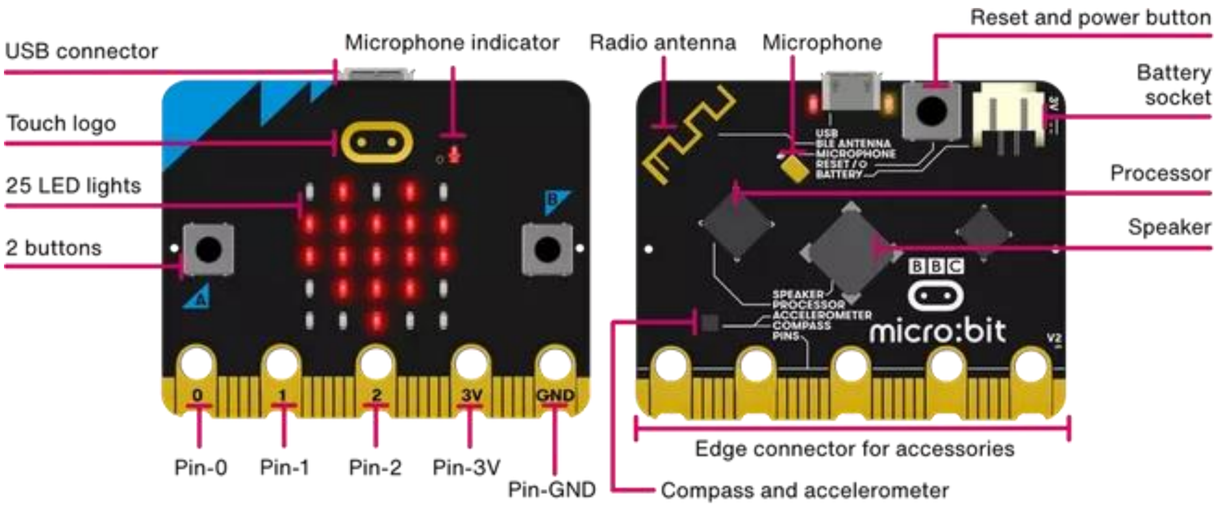


Figure 15: Micro:bit V2 elements description

It is important to know that we can control 3 motors and 3 servos with the multi:bit control board but each pin of the micro:bit can control simultaneously only three activators. So each concept will have the possibilities to only simultaneously use these three activators to move all the parts. In addition, we try to use the same activators for each concept so that we can use the same connector on the multi:bit control board and the same code on the micro:bit. It makes the building process simpler and it reduces the number of expensive parts and use of non-ecological materials.

Figure 16: Battery

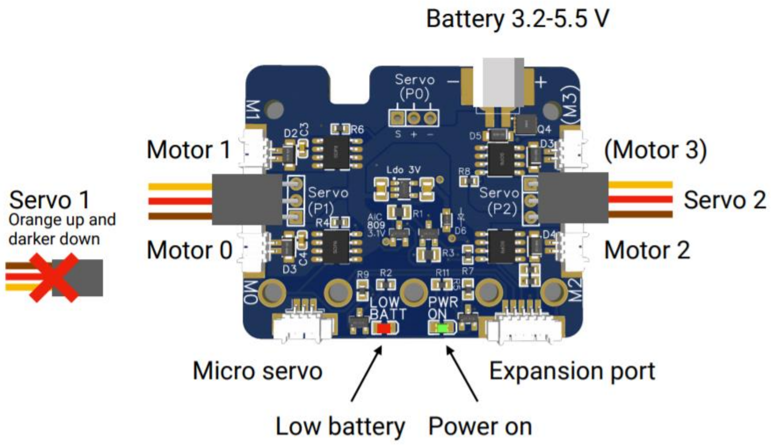


Figure 17: Multi:bit control board

3.4. First concept: the elephant

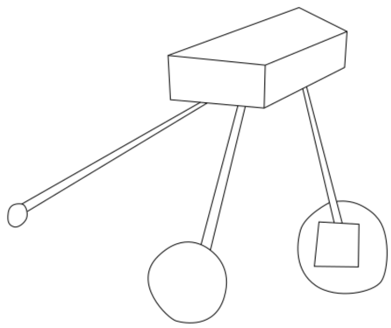


Figure 18: Concept 1

Our first idea was to make something different from what already exists in robot sumo. We usually find robots which look like cars, with four wheels and rectangular form. So we decided to focus on something tall with big legs, so it could tower over its enemies and attack them with a weapon attached to its bottom. We kept the idea of putting two big wheels in the front and one little in the back to stabilize everything. The micro:bit was placed on the top to give a sensation of protection. Having tall structure also could allow the possibility of smaller concepts to pass under and try to kick the legs.

Three wheels configuration is a really good way to create simple robots with good agility and easy controls for young children with funny creations. Moreover, it is easy to push it on its side compared to a four-wheel system which is more stable. This creates a more challenging and strategic game. To move the robot, we found two solutions:

- Using one activator to go forward and backward by rotating the big wheel and one other to change the angle of the third wheel so it can go left or right. This system is similar to a car with a motor and a steering wheel. It can be easier to control because it is what we usually find on car toys but it would be more difficult to build it for our concept. This solution will require at least one motor or a continuous servo and one micro servo to work.
- The second solution was to use a different activator on each large wheel and leave the third wheel completely free. By independently controlling each motor, you can go forward or backward if the two wheels have the same speed or turn by reducing or accelerating the wheel. This solution is probably the hardest to code because all movements are bounded. But it also gives a good dynamic, robots can easily turn on themselves for example. This solution will require two continuous servos to work or to motors.

Because we want to reuse the parts for all concepts, we used the same solutions for all three. We quickly came to the conclusion that using motors was too complicated because of its speed. To reduce the speed, we have to create a gearbox which is a complex part. Moreover, our motors can only turn in one direction.

Continuous servos are a better solution, they include motors and a reduction system and are sold with one compatible wheel. The max rotation speed is 110 RPM but without load. This permits the robot to go up to 1,24 km/h or 34 cm/s. This is low speed but enough for battle robots because the arena usually goes from 77cm to 154cm for common robots. The second solution seems to be the best, we use the same components and the use of two servos instead of one allows the robots to go faster.

3.5. Second concept: the droid

We wanted the second concept to fulfil two requirements, to be the opposite of the first one and reuse at least the base frame and the legs. Because the first one was tall and large, we decided to make this one, low and long. We brought the big wheels closer and put the freewheel far away by using the legs as extensions.

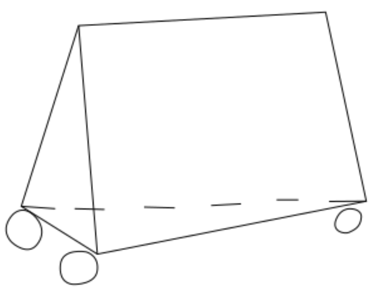


Figure 19: Concept 2

This design normally makes it very stable and strong in pushing or turning the opponent but vulnerable for hits from the side. For this reason, we quickly thought of using a kind of shovel on the front as equipment. It can be made by using a piece of cardboard and a cardboard holder that already exists in the Hover:bit from MakeKit. It is used with one micro servo. This system can also allow the possibility of children to adapt equipment by changing the form of cardboard or putting something else into the cardboard holder.

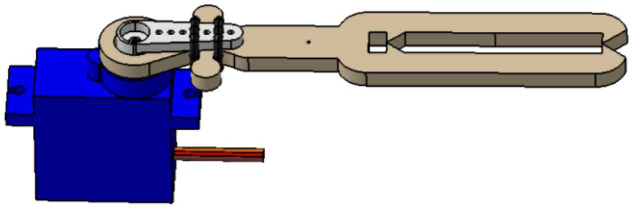


Figure 20: Cardboard holder

3.6. Third concept: the scorpion

For the last one, we tried to change the dynamic of the robot. We put the driving wheel in the center to give the best agility compared to the over. It will also be easier to control. In addition, we installed two freewheel this time to create something symmetric and add more stability. Compared to the others, this concept is less stable but more agile so it can move easily around the other to hit them from the side.

We also decided to put freewheels higher than driving one. This adds a funny balancing movement when you accelerate or brake. In order to have this movement, we need to have a good centre of gravity so that the robot can go to both sides. By playing with this system, it is possible to hit the opponent at different heights for example.

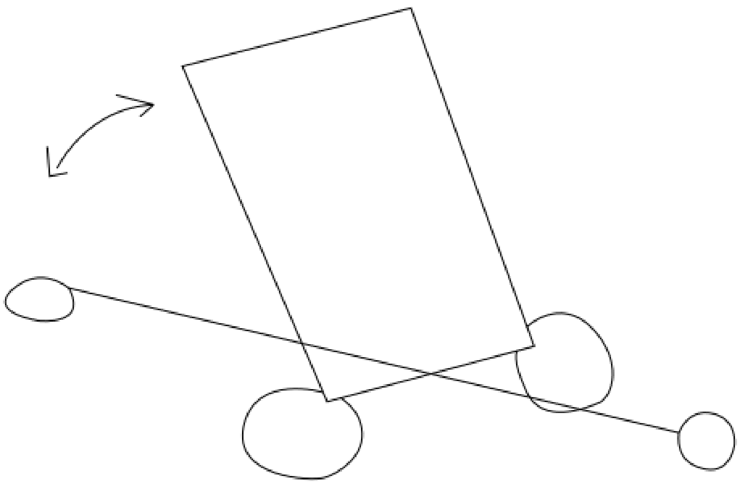


Figure 21: Concept 3

4. Design

To develop our three different concepts, we used an iterative process. First we created the design on a 3D modelling software and shared it with MakeKit to have feedback. Then we made a real prototype in the Makerspace to test the product. We were able to see problems and improve them by doing the same process over again. When our objectives were satisfied for the current step, we could move on to the next one.

We have chosen to divide the robot’s design in three steps:

- The first step was to have a working wood skeleton of the robots that can be assembled and moved with the driving wheel.

- The second step was to complete the skeleton with equipment for each concept and test it against the other. That was the hardest part because all the robots are different and it is difficult to have something that can be efficient at the same time against two different concepts.

- The last step was to create a cardboard shape that can be put on the skeleton. One idea was to create an animal shape for each one and give the possibility to the children to paint it or draw on it and have a custom robot.

4.1. Skeleton

4.1.1. 3D modelling software

We have chosen to use two software tools to design robot:

- CATIA: It is a multi-platform software suite for computer-aided design (CAD), computer-aided manufacturing (CAM), computer-aided engineering (CAE), PLM and 3D, developed by the French company *Dassault Systèmes*. We used it for all the design phases because it was the software where we had the most experience. It is also very complete with simulation, conception, drawing and the possibility to update all parts very easily with parameters that is really interesting when your development is based on an iterative process.
- Fusion 360: It is also a multi-platform software developed by *Autodesk*. It is easy to learn but there are less possibilities than CATIA. It is the software used by MakeKit and we only used it to transfer our 3D parts from CATIA to Fusion 360.

The advantages to using a 3D modelling software is that we can simulate the robot to know if we can assemble it or if we have collisions between parts. It is also a good way to show prototypes during meetings and have the possibility to test modifications in a few seconds without using materials. In addition, we used CATIA to directly export all the parts into a file for laser cutting.

4.1.2. Fixation systems

The robot’s skeleton is composed of wooden parts that need to be connected. For this we only used non-permanents links like screws, O-ring or interlocking systems.
We tried to mainly use the “MakeKit system” connector to connect two wooden parts. This system uses male and female connectors with a rectangular form. To have a good connection between two parts, we only need to control the width and the length of connectors because the wooden plate thickness can vary between 3 mm and 3.6 mm. However we know that this kind of fixation is perfect for positioning parts but not always for maintaining them and we need to secure it with O-rings or screws.

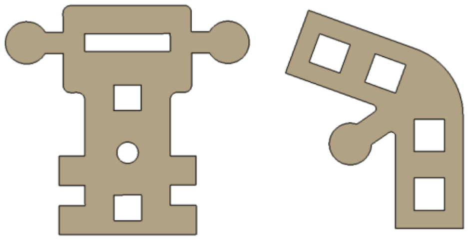


Figure 23: Male / Female

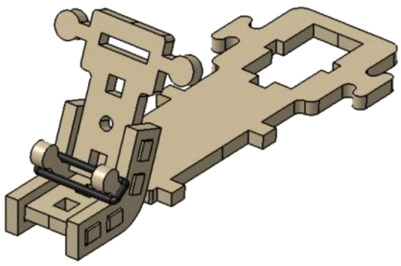


Figure 22: "MakeKit's system" connector secured with an O-ring

Then we have used O-rings to fix objects on the skeleton such as servo motors, pens, batteries and to secure the MakeKit "system" connector. To be able to fix these O-rings on wooden frames, we added an ear system on either side of the object to be fixed. We had the possibilities to used two sizes of O-ring which have an influence on the size of the ears:

- Small silicone O-rings 7mm inner diameter x1.5mm thick, used to fix servo on cardboard holder and to secure angle connectors (Figure 22).

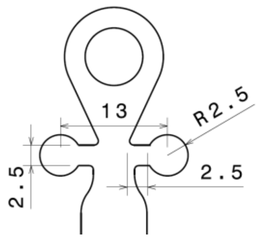


Figure 25: Ears specifications for small O-rings

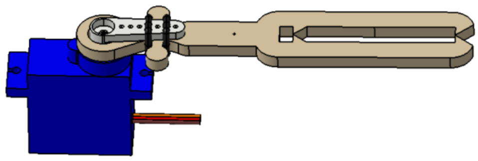


Figure 24: Small O-ring usage example

- Large silicone O-rings 28mm inner diameter x2.5mm thick, used for the other things.

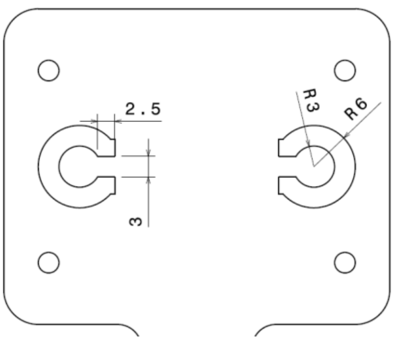


Figure 26: Normal ears specifications

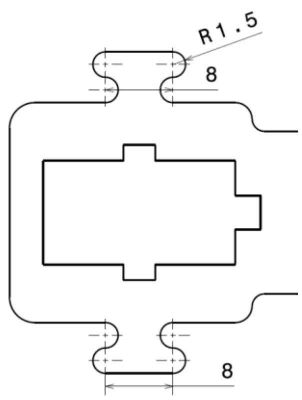


Figure 27: Ears specification for wider objects

The last way to fix parts is to use nylon screws and nylon nuts. They all are 3 mm diameters and we can choose between several lengths such as 12 mm, 16 mm and 20 mm. They are mainly used to fix the micro:bit on the control board or to make our freewheel. We have also created wood spacers that can be used with these screws.

Figure 28: Nylon nuts

Figure 29: Nylon screw

Figure 30: Spacer

4.1.3. How to fix servos motors on wood?

We have chosen to use the same system for the Hover:bit with hole for positioning and O-ring for maintaining. However, this system does not allow us to add wheels on our robot so we had to make a new system using the same characteristic to be able to fix servo motors in a different position.

Figure 31: Servo fixation 1st solution

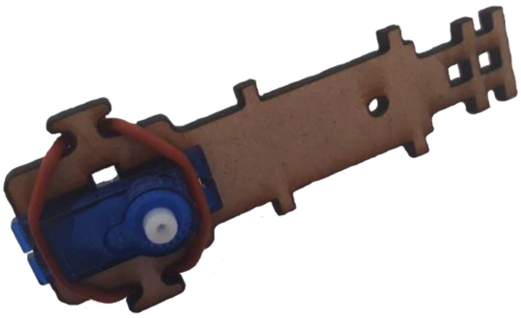


Figure 32: Servo fixation 2nd solution

During our experimentation we had two problems to place servo motors:

- We had not thought of leaving a space to pass the cable so it was impossible to put servo motors in place.
- Depending on the servo’s type, holes were sometimes too small or too large so it was impossible to put servo motors in place or they weren't secured by the O-ring. In addition, we found that the dimensions given in the data sheet were not always correct. So we decided to measure each servo to find out the real dimensions and if there were any differences between the same two servos.

In figure 31 we can find the actual dimension for the continuous servo and the micro servo. We are only interested in the **B** and **F** dimensions for our system and we notice that the dimensions only vary by 0.1 mm. We have therefore chosen to keep the larger dimensions. Furthermore, we can see that the other dimensions seem to be accurate with small differences. Finally, we can note that the micro servo is 1 mm shorter.

Continous servo	1	2	3
A	**30.5**	30.5	30.5
B	**22.8**	22.8	22.8
C	**27.7**	27.7	27.7
D	**12.3**	12.3	12.3
E	**32.2**	32.2	32.2
F	**16.7**	16.7	16.7

Micro servo	1	2	3	4
A	29.7	29.7	29.8	29.8
B	22.7	22.7	22.7	22.6
C	26.7	26.6	26.7	26.7
D	12.3	12.3	12.3	12.2
E	32.2	32.2	32.2	32.2
F	15.7	15.6	15.8	15.6

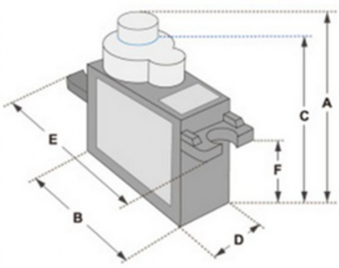


Figure 33: Continuous servos and micro servos measurement

4.1.4. “Micro:bit card” principle

Our robot is controlled by a microcomputer called micro:bit. It needs a battery and a control board to run and activate the motors. All the designs can be controlled with the same code but we want to give the children the possibility to improve or adapt it for themselves, for example by changing the sensitivity or adding features. We also want to give them the possibility to share these programs with other people by simply changing the micro:bit on the robot instead of using a computer to transfer it. That is why we added the idea of creating a "micro:bit card" composed of all the necessary elements for the operation (micro:bit, control board, battery) which can be put or removed on the robot in a few seconds and very easily. As for the USB key, each robot will have a universal attachment that should be present on the part if someone wants to create his own frame.

Our first version of the system was a simple wooden plate with:

- On one side the micro:bit and the control board are held in place with two screws.
- On the other side, the battery is fixed with a large silicone O-ring.

The wooden card was composed of a male connector that works as for connector of the "MakeKit system". This system was really easy to plug it in but was not really secured, especially when the thickness of the wooden card tended toward its lower limits. Moreover the position of the micro:bit card was limited on the base frame and an idea was to have the possibilities to choose its orientation and place to change the design and the gravity center.

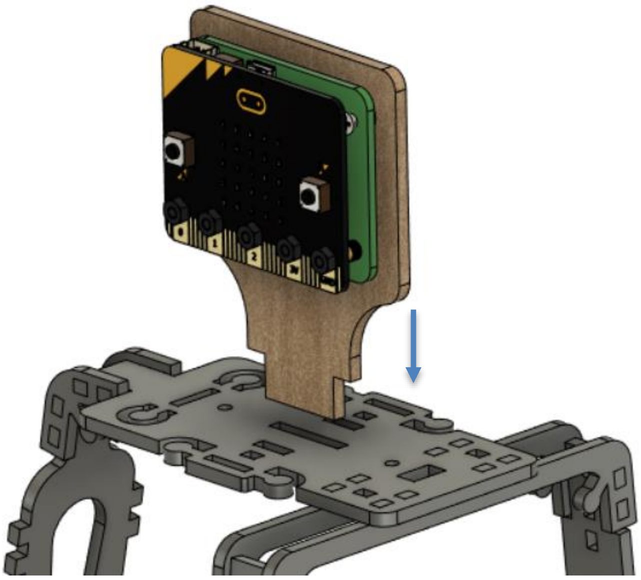


Figure 34: "Micro:bit card" version 1

In the second version, we used a kind of box that can be moved on the base frame. Indeed, we made a lot of holes on the base frame to be able to move it along the length side and we also gave the possibility to rotate it. The micro:bit card was inserted into this box and secured with a pin to avoid it falling during battle.

This system was the best with a lot of possibilities and very good fixations but it also added many parts and raised the complexity. Moreover, assembling the box was too complex for young children compared to the previous system. During testing, some parts tended to break at the base of the box. Finally having a lot of position for the micro:bit card was not relevant because concepts need it at the same place to have a good gravity center and we decided that only having the possibility to rotate it by 90° was sufficient.

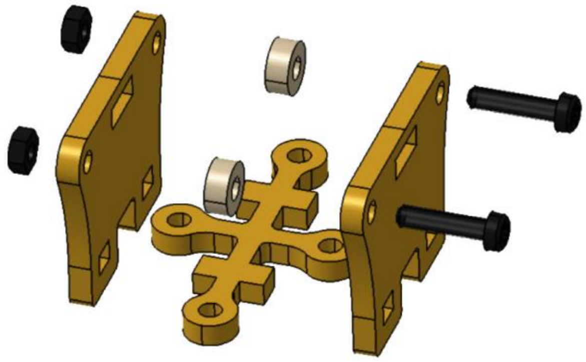


Figure 35: Block assembly

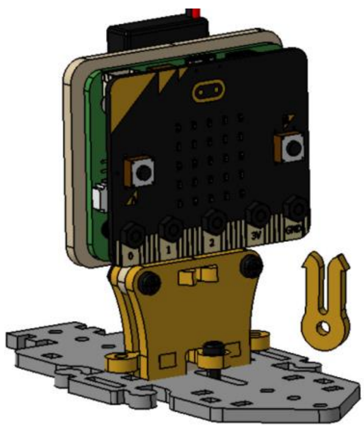


Figure 36: "Micro:bit card" version 2

We found the best solution by mixing the two previous ones:

- We used the same solution to plug in the micro:bit card on the base frame but we made a cross design to be able to rotate it.
- We added a pin to secure it and prevent it from moving too much when the thickness of the wooden card tended toward its lower limits.

At last, this solution remains very simple and ergonomic in its use. It is also very suitable for repetitive use, such as experimenting with Battle:bit code. Moreover, it only adds a small part which will have a low impact on material consumption during laser cutting.

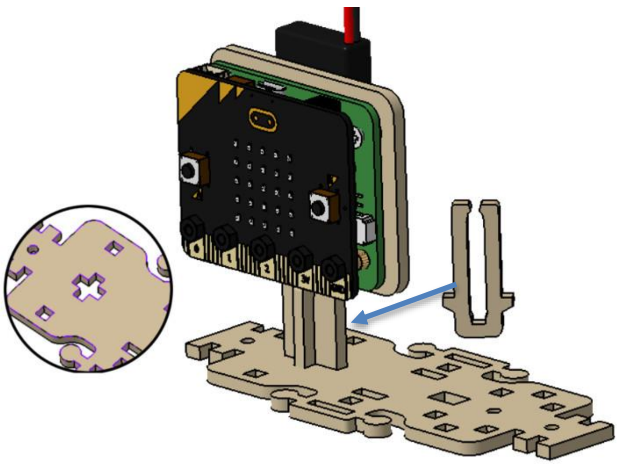


Figure 37: "Micro:bit card" final version

4.1.5. Equipment

The equipment has power over each robot to help it win the battle. We tried to come up with simple ideas that use simple elements like cardboard or a pen. These elements are attached using the cardboard holder and activated by a micro servo or a continuous servo. We tested these equipment on the robots and did battle to determine which one was the best for each robot. We tried to have the same chance to win between all the Battle:bit. Moreover, all these equipment can be modified because they use common materials. The users also have the possibility to create new equipment or to use different activators like motors or solenoid if they want and have access to it. We are offering three robot concepts but the children are totally free to design a new one if they want.

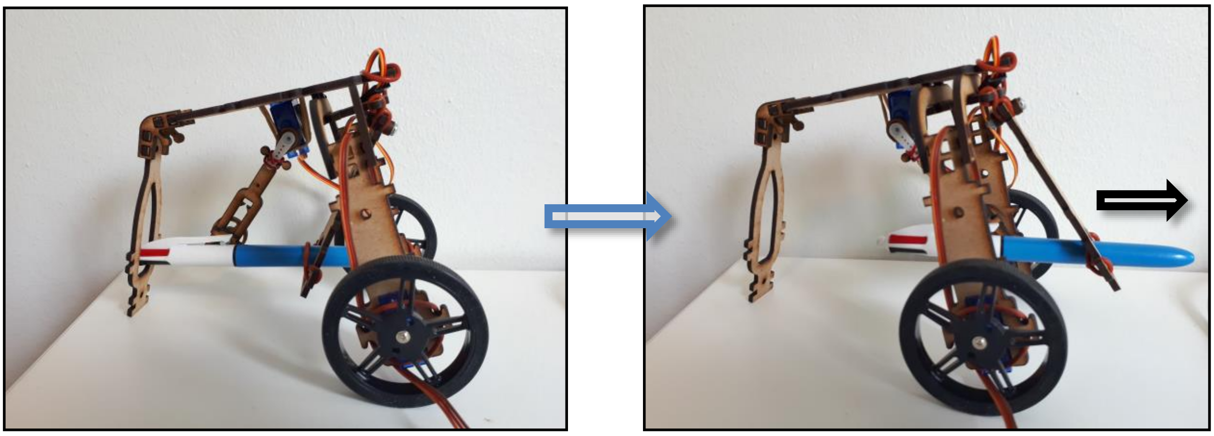


Figure 39: Equipment 1

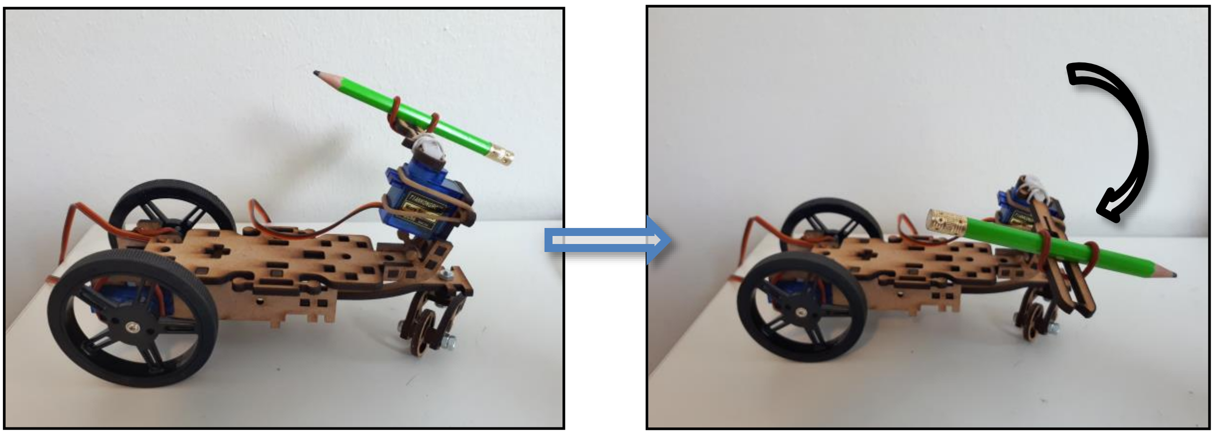


Figure 38: Equipment 2

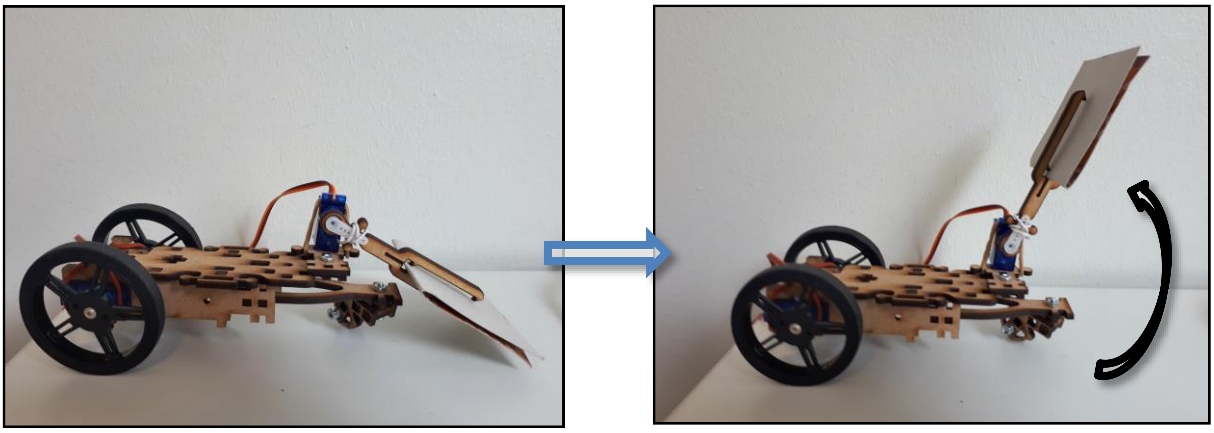


Figure 42: Equipment 3

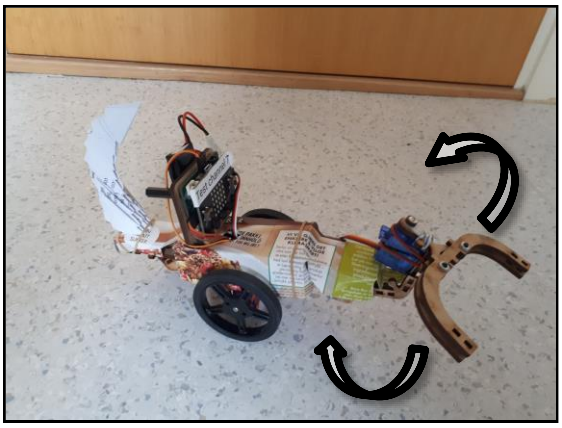


Figure 41: Equipment 4

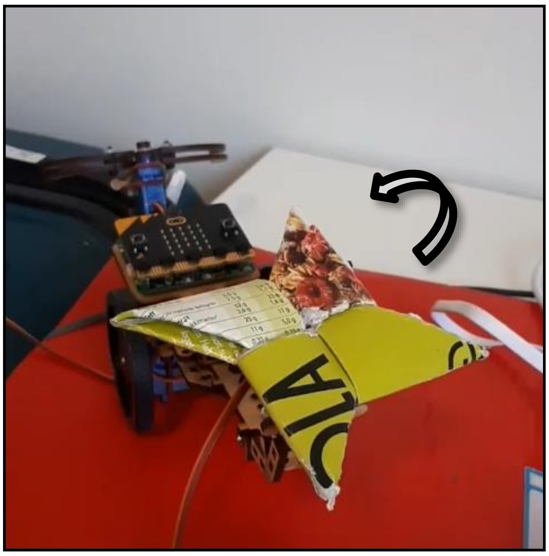


Figure 40: Equipment 5

4.1.6. Evolutions

Our development process was based on an iterative process that can be seen in the evolution of all the wooden parts. We can see that the main parts had 5 significant stages of evolutions:

- The first version was very functional with only the necessary elements to allow the construction and testing of robot dynamics.

- The second version was quite similar to the first. We adjusted all the dimensions and tried to solve all the problems such as adding space for the servo cable or adding material around the holes to avoid breakage.

- The third version was the biggest improvement with many goals. The first one was to improve the rendering of parts to make robots prettier by putting more round corners for example. We can see it in particular on the large connector. The second one was to find the best configuration for each robot to have a good dynamic and fair battle. We already had functional concepts at this point with the first equipment version. To test this different configuration, we decided to add a way to attach components to the main parts. We could then change the location of the freewheel, gear or micro:bit card to change parameters such as stability, agility or power without the need to build new parts.

- The fourth version was intended to make the assembly less complex and to increase the modularity, by using the same fixations on a maximum of parts and using symmetry. We also implemented the last version of the micro:bit card.

- Finally, the latest version has mainly removed the remaining unnecessary holes used in the third version

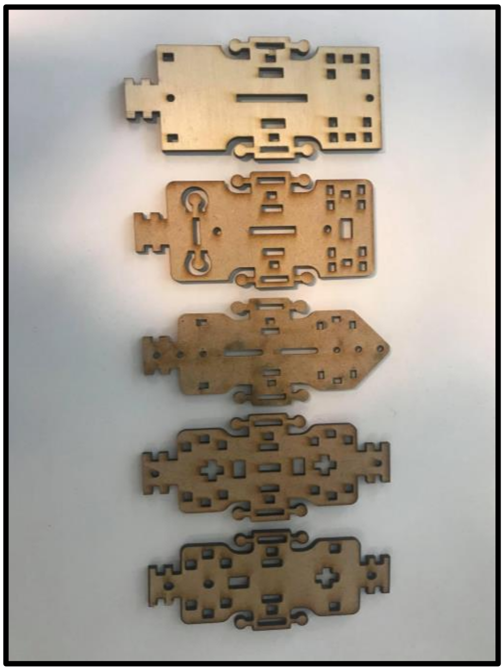


Figure 45: Base frame evolution

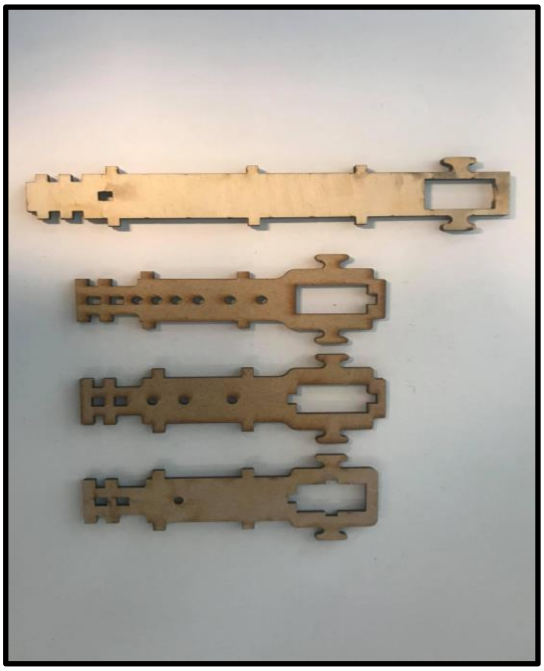


Figure 44: Servo holder evolution (version 2 is missing)

Figure 46: Third leg evolution

Figure 43: Large connector evolution

4.2. Cardboard body

4.2.1. Mission

Our robots are made of a wooden skeleton and we decided to create a cardboard body on top of this structure. With this element, Battle:bit users will be able to add cardboard shapes such as animals to create a fun new robot.

4.2.2. How to fit on the wooden skeleton?

The idea is to create a cardboard piece that fits over the wooden skeleton. But the question arose how to best attach these two elements.

The first step was to think about how the skeleton was made, how the different parts were put together so that the cardboard could be attached. With the base frame and its shape and the micro:bit base frame, it was possible to create a cardboard shape that stayed attached to the robot.

It was then necessary to know all the measurements of our robots and all the parts that make them up in order to start drawing cardboard shapes.

4.2.3. Steps of design

- **Drawing designs**

We wanted to create a shape that could be assembled without the use of glue through a system of notches and slots to meet our goal of sustainability.

- **2D modelling software**

Inkscape is a free professional quality vector drawing software. It is used by professional and amateur designers around the world to create a wide variety of graphics such as illustrations, icons, logos, diagrams, maps and web rendering.

In order to be able to cut out the cardboard shape and to print it as a template, we had to draw it in two dimensions using the Inkscape software.

- **Test and improvement of the designs**

Thanks to the tests we were able to do at the Makerspace, cutting, assembling the cardboard and testing it on the wooden skeleton. A few improvements were necessary to ensure that the body would stay on the frame and to achieve the final shape.

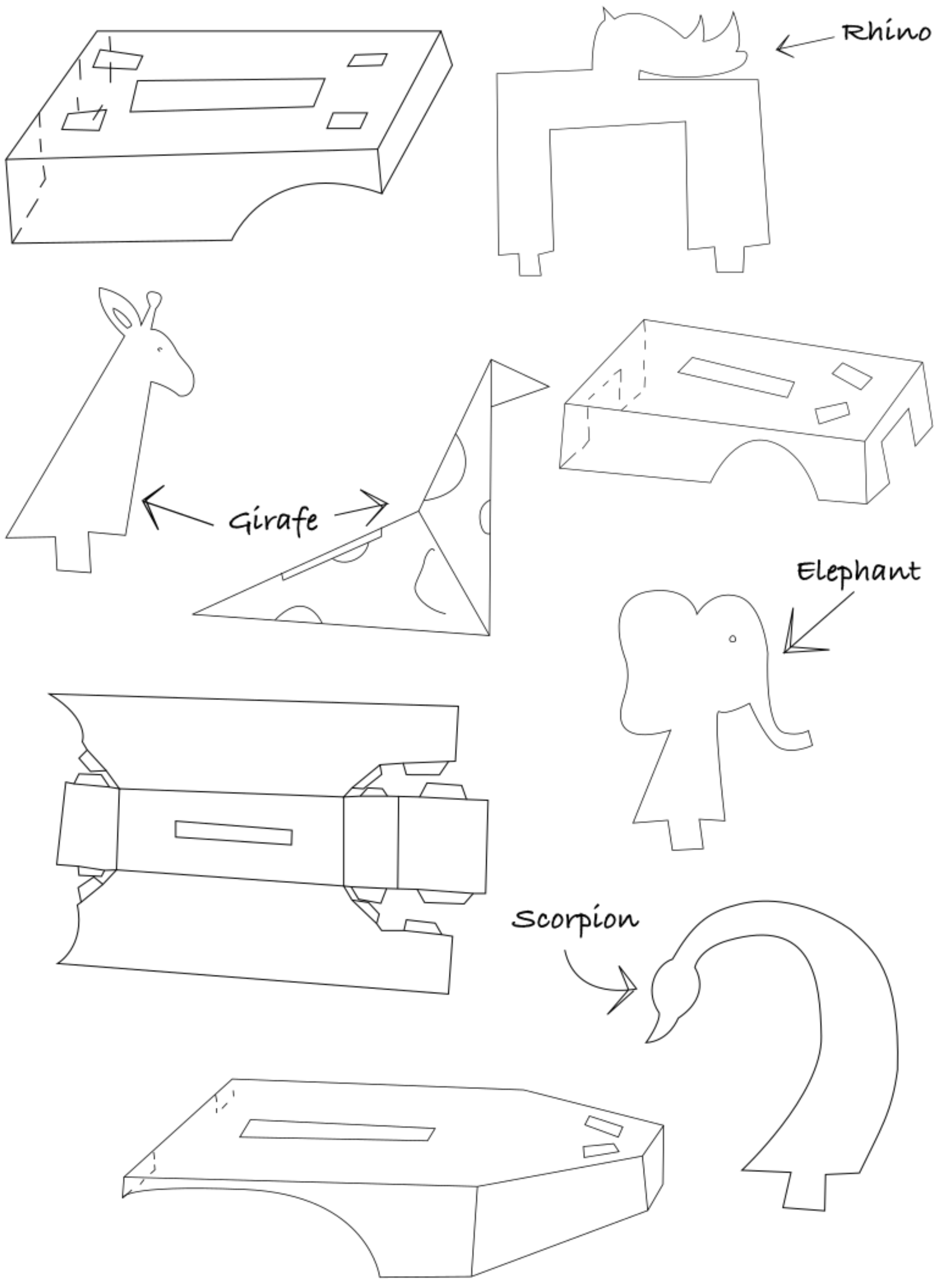


Figure 47: Animal's shape ideas

5. Codes & Instructions

In this chapter, we talk about aspects of the project that permits to use the kit and control the robots. We explained how we code the micro:bit which is used to move the robots and to control it. We also talk about diverse solutions that allow us to control the robots by using devices like computers or smartphones instead of using a second micro:bit. Finally, we explain how we developed instructions which make the kit accessible to elementary school children.

5.1. Makecode

MakeCode is an open source platform to create an interactive and engaging experience for those new to programming. It offers users to program visually in JavaScript or Python. With this they don’t have to learn the correct syntax of the coding languages but only touch on the logic behind it, which gives beginners an easier gateway into textual programming. Makecode also has the option to switch to textual code to see how the code you created visually with block code would have been written. Furthermore they have a simulation option which means the students would not necessarily need their own micro:bit to learn.

This is the platform we use to program the micro:bit. Both the micro:bit on the robot design as well as the controller.

For the controller we use radio signals to send values. Each value means something specific like moving forward, turning and activating the arm. The robot itself will receive radio signals and do specific things depending on what it has received. For movement and the armed servo we’re using booleans, which means only sending 1 and 0 or true and false. Once the movement is activated there will be values sent depending on how much you are rotating the controller to turn.

Please go to appendix p 46 to see the complete code.

5.2. Android App

To make the robot's more accessible we thought of creating a mobile application that controls the robot. This will result in us only needing one micro:bit in the kit as the user can use their smartphone instead. MakeKit already worked on an IOS app but did not have one for an Android device. We began the development of this app but it is unfinished due to time restrictions. However the most difficult part of the code is complete, which is the Bluetooth connection and communication between an Android device and the micro:bit. It still requires some fine tuning to send the values that control the robot.

5.3. Progressive Web App

A progressive Web App is a more advanced version of a website that appears as a smartphone application for the user. In other words they should have the same features and look as a native app does. Progressive Web Apps or PWA for short are accessible by either visiting a given URL or by installing the PWA directly to your device. A Web App is usable offline when installed. In comparison, a traditional website can only be accessed through a web browser. The reason we're looking to use the progressive web technology is for ease of use, so that no matter what device is being used it is easily accessible by all students to control the MakeKit robot. The fact that the app needs to be accessible with any type of device means you have to keep in mind that the content of the website must adjust accordingly.

There are several advantages to using a Progressive Web App over a conventional web app. For starters, a big upside is that it allows notifications as if it was a native app. Which means you do not need to open the website to give information to the user and keep them interacted. PWA's are installable on smartphones and PC's. This causes you to have the possibility of using it offline and seeing it in your personal installation directory. Whereas traditional apps you can only have a bookmark on the browser. Since the app is accessible through your browser, updates are not required for the user. You can reserve any push notifications for engaging the user and not bother them with update permission requests. Another advantage is that Progressive Web Applications must be hosted over HTTPS to make sure the data is being delivered securely. Therefore, any interactions are as safe as any secure website using https would be. But one of the most unique advantages is that A PWA allows web developers to create a mobile application with languages they already have experience with, opening opportunities to develop mobile apps to many people.

5.4. Instructions

The final product needed to be explained in Instructions, so Students and Hobbyists could easily build it at home.

In our first step where we had to optimize and simplify the building process, we were able to use our experience from assembling and disassembling the Battle:bit in every development iteration. During our final build we took simple photos with a phone of every step to document it and to later use them as visualisation in the Instructions.

The second step was to think of a concept to structure the Instructions and MakeKit provided us other manuals of theirs to adapt their typical structure, and they gave us feedback for what is especially important for them to be included.

Very important for them was having a list of all the content of the Battle:bit box and the exact amount of all the parts, as well as the amount of the parts for every individual build and in every individual building step.
Another thing we did for every individual build was determining which building blocks they have in common, so we do not have any unnecessary repetitions. In the first draft we used the photos we took in our final build to depict every building step.

6. Results

In this chapter we present the parts we created to make the Battle:bit kit. We focus mainly on the wooden parts and the cardboard body parts. We also discuss the current state of the progressive WebApp and of the instructions.

The kit parts are finished but remain prototypes that can be used by MakeKit to create a new product based on our project. Indeed, we need to convert our files so that they can be used to produce the parts on the MakeKit laser cutter.

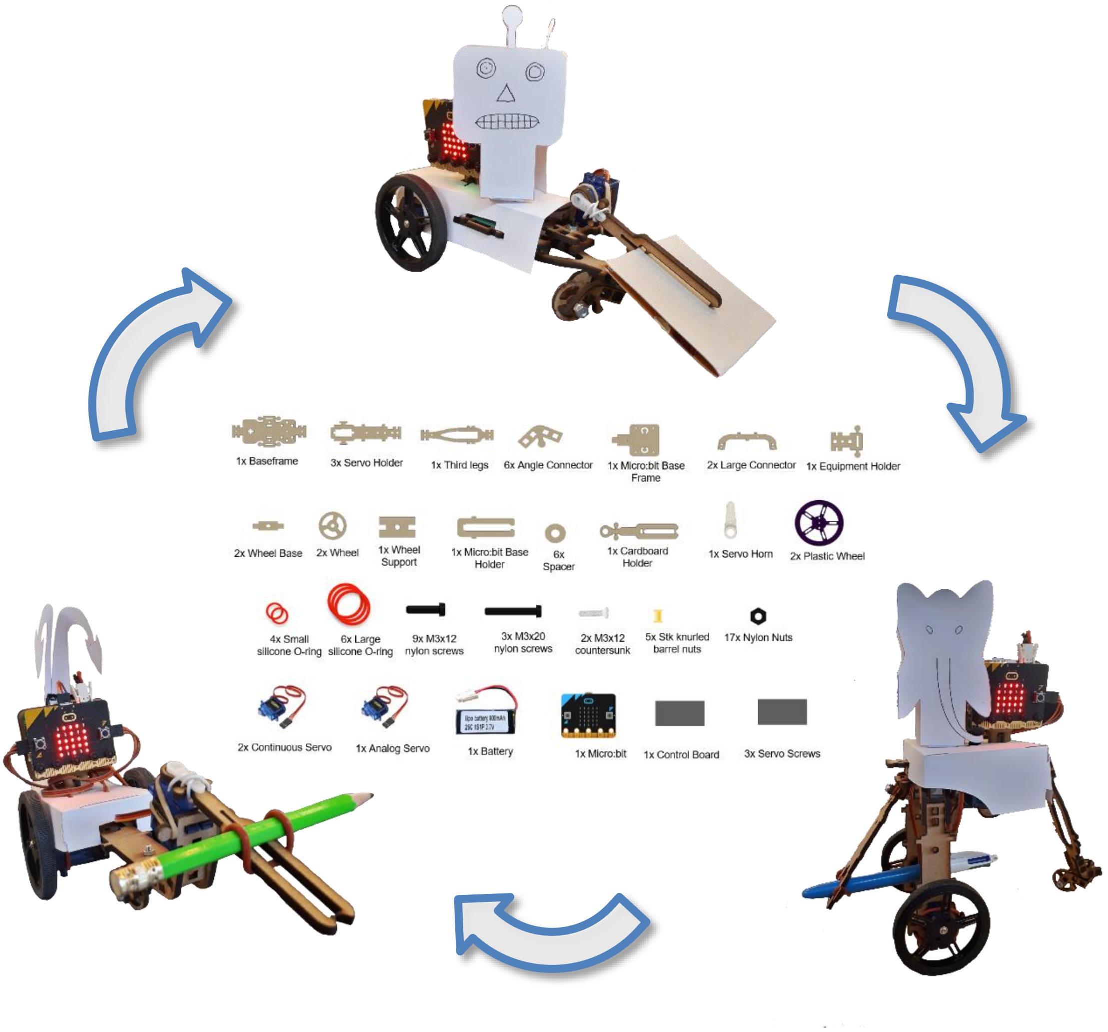


Figure 48: Battle:bit kits possibilities

6.1. The wood skeleton

We have created 13 wooden parts to build the skeleton. The part's quantity can be different between the concepts so we have chosen to put in the kit the maximum quantity needed for all the parts.

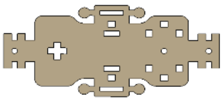
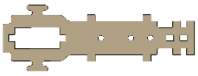
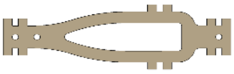
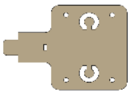
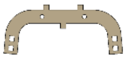
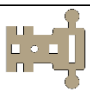
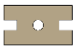
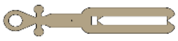
Name	Picture	Quantity for one kit	Quantity for concept 1	Quantity for concept 2	Quantity for concept 3
Base frame		**1**	1	1	1
Servo holder		**3**	3	2	2
Third legs		**1**	1	1	0
Angle connector		**6**	4	2	6
Micro:bit base frame		**1**	1	1	1
Large connector		**2**	2	0	0
Equipment holder		**1**	1	1	1
Wheel base		**2**	2	1	2
Wheel		**2**	2	1	2
Wheel support		**1**	1	0	0
Micro:bit base holder		**1**	1	1	1
Cardboard holder		**1**	1	1	1
Spacer		**6**	5	4	6

Figure 49: Wooden parts inventory

6.2. The cardboard body

The cardboard part is composed of two elements, a base fix on the wooden skeleton and a decorative shape.

Each robot has a specific base which fits perfectly with its skeleton and it is possible to add the decorative shape chosen by the user. All the cardboard pieces can be constructed and assembled without glue thanks to a system of notches and slots.

The red lines should be cut and the blue lines should be folded by the user.

- The cardboard bases:

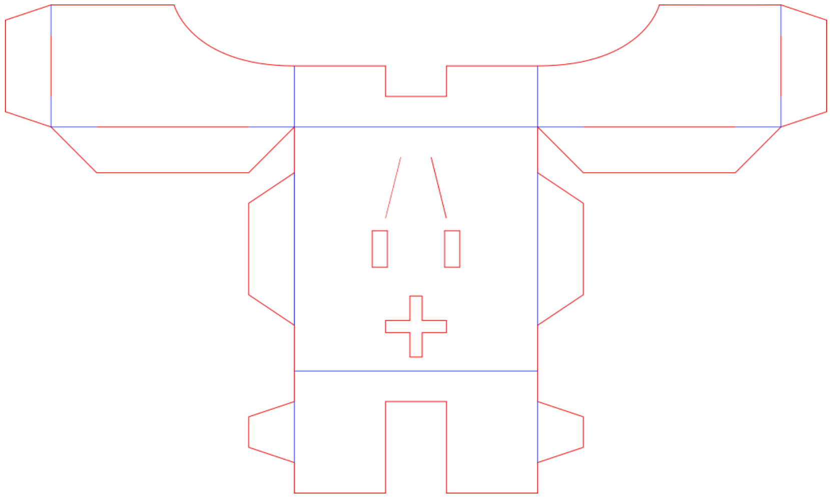


Figure 52: Cardboard base for concept 1

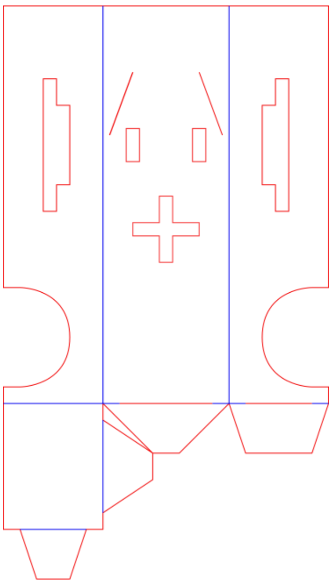


Figure 50: Cardboard base for concept 2

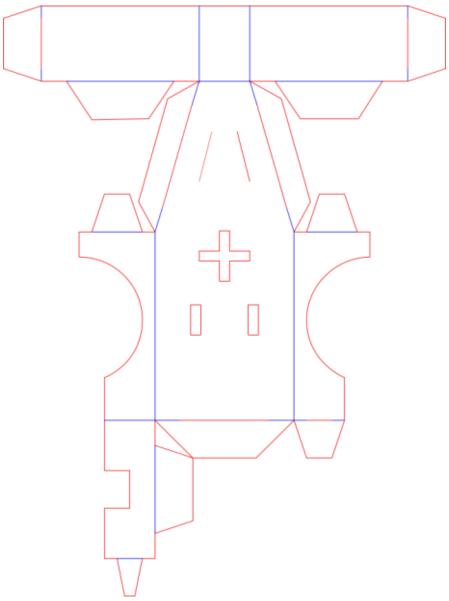


Figure 51: Cardboard base for concept 3

- The decoration shapes:

Each decoration shape has been created to be assembled with a specific robot, as the skeleton shapes reminded us of animals, but the user will be free to choose their favourite element or create their own using the same folding and splitting method.

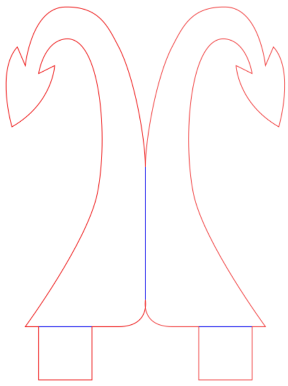


Figure 53: Scorpion tail

This cardboard **scorpion** tail was created for the balance robot because the way it moves and the shape with the weapon in front reminded us of a scorpion. We decided to create a tail to put on the back of the robot.

This robot is also fast and can turn very easily like a scorpion.

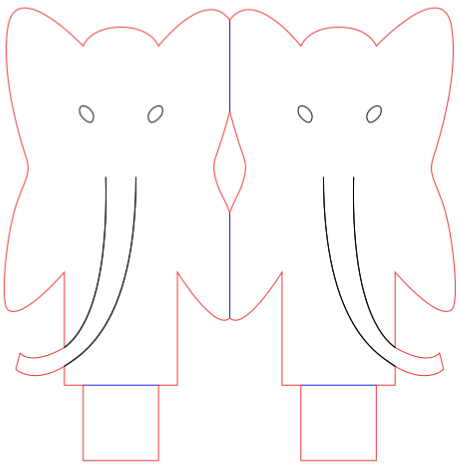


Figure 54: Elephant face

The **elephant** reminds us of the long legs robot (concept 1) because of its size and its weapon that can swing like an elephant's trunk.

This robot is also slower to turn and can fall over if the user makes too sudden movements. An elephant will also live by moving forward, but it has more difficulty turning around and moving sideward and backward.

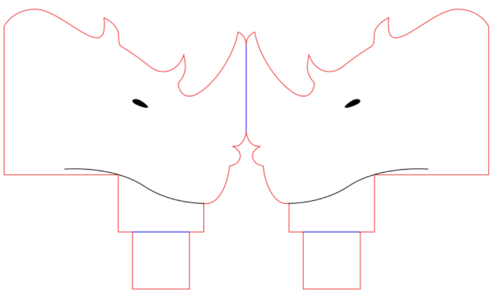


Figure 55: Rhinoceros face

The **rhinoceros** is for the concept 2 because of its weapon that can knock down other robots by lifting them up.

On the advice of MakeKit, we created a final decoration that looks like a **droid** because that is what our robots are in the first place. We also used it for the concept 2.

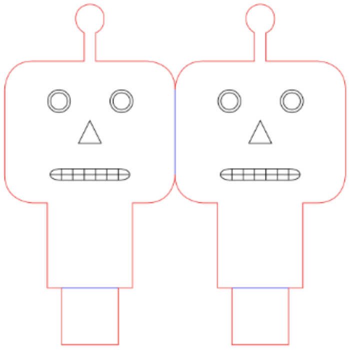


Figure 56: Robot face

- The cross:

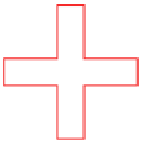


Figure 57: Cross design

To enable the micro:bit base frame and the base frame to be joined together, we have created a cross on the cardboard form. This element also allows the cardboard to be attached to the wooden skeleton.

- The lines:

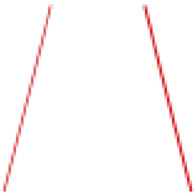


Figure 58: Lines design

These two lines are slots and they allow the decorative form and the cardboard base to be assembled by simply folding the bottom part of the decoration and pushing it into the slot.

6.3. The Web App

The apps still need some work to finish it. These applications were taken in addition to the original project mission.
Bluetooth communication is possible in the android app. You are able to see your micro:bit in a list and connect to it. The next step for the android app is creating a user interface to control the robot. The buttons in this UI will send values which the robot understands as a specific type of movement. This also comes with the changes to the MakeCode. The example code we made for the micro:bit must be changed since we are using radio signals between the robot micro:bit and the controller micro:bit. The code needs to be changed from using radio to using Bluetooth.

Same goes for the Web App. We have begun the code of scanning for any Bluetooth devices around. However we're unable to actually see any devices. It is important to be using a browser which supports Bluetooth communication. Newer versions of google chrome, Microsoft Edge and Opera work. As an example for Google Chrome you will also have to allow, "Bluetooth device permissions" feature in the google chrome settings by navigating to "`chrome://flags/`". The issue of no devices found when scanning is unresolved.

6.4. Instructions

For the final version of the instructions we at first replaced the lower quality photos with rendered pictures of Thomas' 3D build of the robots, and added graphics, like arrows and highlights, to better communicate motions and directions.
The biggest challenge for us was to write the explanatory texts so children of different ages can follow it, especially because none of us has any educational experience. But we decided to focus more on delivering something to MakeKit that they can work with, when we are no longer involved in the project, knowing that they will have to polish the final product anyway. That is why in the end we, even though we wanted to write as less descriptive text as possible and mainly let the graphics speak, still went into more details than necessary. Therefore, we would call it a final draft rather than finished.
Now it includes introductions to MakeKit, the micro:bit and the Battle:bit, as well as a list of content and instructions for building the three different versions of the robot, the coding of it and the controller and for handcrafting the cardboard body, and it concludes with the rules on how the robot battles work.

7. Conclusion

During this semester of the European project and thanks to the collaboration with MakeKit, our team was able to meet the project demand by creating a Do It Yourself Battle:bit using mainly wood and cardboard. It is a new competitive item in the kit market, as it does not use plastic parts and addresses the issue of sustainability.

Throughout the project, the team dealt with the problems they faced by communicating, holding weekly meetings with the company and their supervisor and following the schedule set at the beginning of the project to deliver a finished product.
We all learned a lot during these months by working in a multidisciplinary and international team. We now know how to make the most of each other's knowledge and skills, and this has enabled us to produce a more sophisticated product than originally planned, as each of us has been able to add our own personal touch.

Through this project we also learned about the machines and how to find processes that are environmentally friendly and more suitable for small businesses. We had to think about the production and tools available at MakeKit but also about the process of building the kit for children.
Our kit adds new possibilities for the children and in the creation process as they have to choose the robot they want to build and they can develop their creativity by choosing the cardboard body and adding the decoration element they want. They can also create new decorations or draw on it as it is white cardboard.

Although we have delivered a finished product to the company, the project is still ongoing and has allowed MakeKit to expand their offering. They are looking for interns to develop an application to connect a phone to the micro:bit and control it.

8. Appendices

8.1. Links

Makecode: `https://makecode.microbit.org/`

Controller code: `https://makecode.microbit.org/_8UuJUhe3t89K`

Robot code: `https://makecode.microbit.org/_Eyg4rw3kJYjv`

MakeKit website: `https://www.makekit.no/`

Documentation: `https://www.makekit.no/docs`

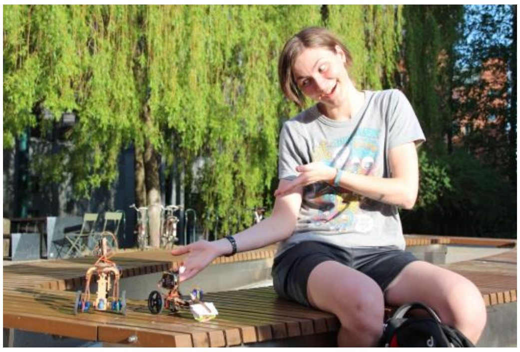
8.2. Team

We are a team of four international students from all over Europe, with different backgrounds and knowledge.

Kasper Ruys studies Electronics-ICT. He has seen the basics of electronics, meaning he has made electronic circuits, printed PCB's and has had multiple projects involving, Internet of Things. He has more experience in programming and app development as he created multiple applications for smartphones.

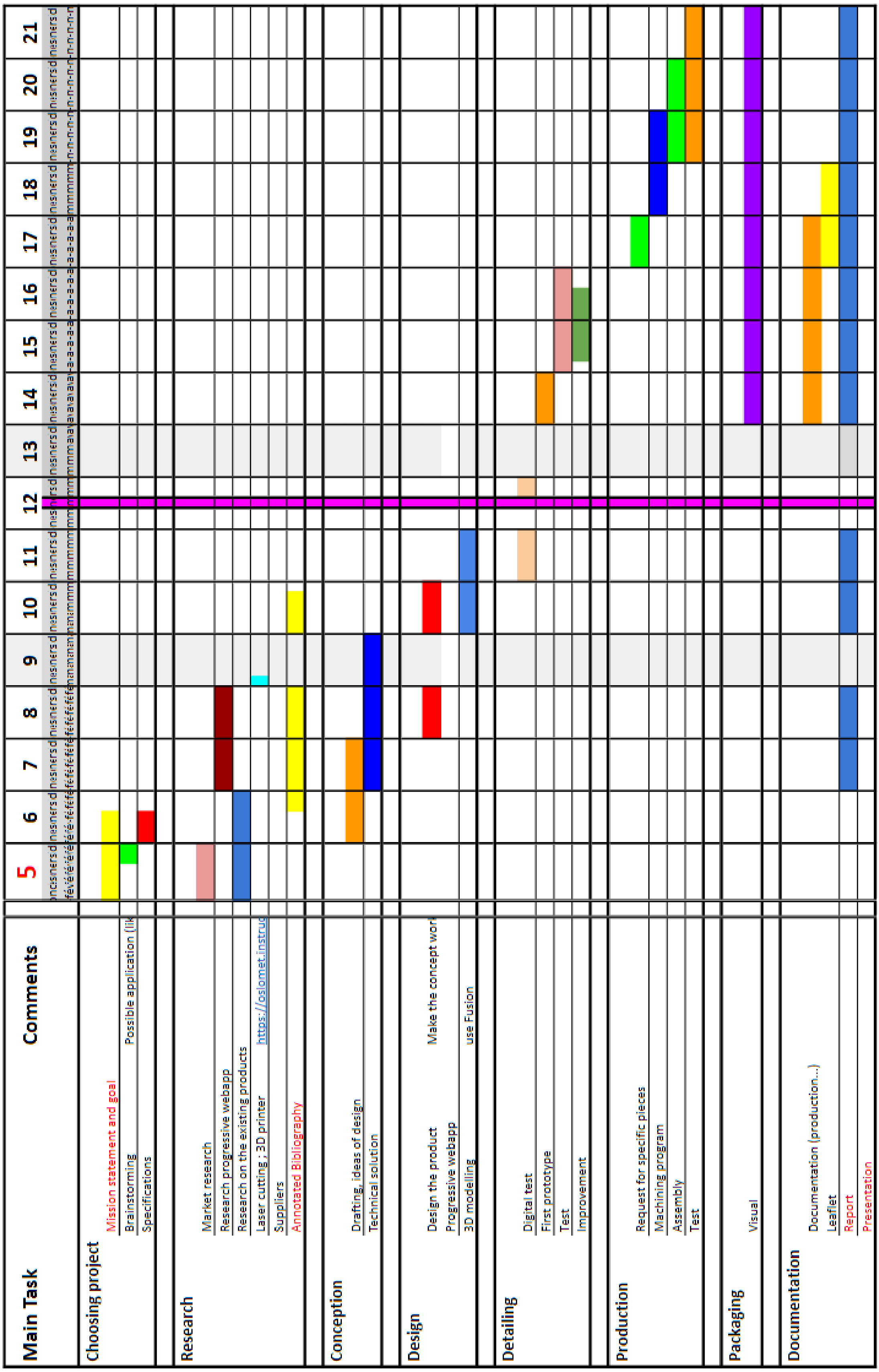
Thomas Combes studies mechanical engineering. He is good at finding technical solutions and making things work. He handles the design phase of the product. He also has knowledge of materials and machining.

Julie Mallard is a student in an engineering school specializing in packaging. She studies materials, interactions between product and packaging, design, general sciences, eco-conception, marketing. Her design skills and knowledge of materials are her main asset for this project.



Verena Hanner studies Interactive Media with a focus on computer science. She has experience with film and animation, as well as Interface Design, Computer graphics and Game Programming.

8.3. Project plan



8.4. Stakeholders analysis

- Our team (Core stakeholder)

We are the main actor because we are the decision-makers and are going to carry out the whole project.
Therefore, we will gather practical experience and learn to properly work in a team. As well as working with a company as a project partner which will give us insight in real production processes.
This project allows us to broaden our knowledge by working on different subjects that we are used to in our own countries.

- Steinar Holøs and Henning Pedersen (MakeKit) (Primary stakeholder)

Steinar is the CEO and Henning is the CPO of MakeKit. They are the cooperation partners of the project and plan to use the concepts and products developed in this study as a real product for their MakeKit Line.
Therefore, every design decision has to be discussed and agreed with them. Their company values should influence the design process as well. All of that has to be considered and brought in line with the team's ideas and interests.
They give us support and supply us with helpful advice as well as materials. And they are the proprietor of the project, they have to be informed all the time of the progress of the project and we need to share with them all the information.

- Alfredo Carella, our supervisor (Primary stakeholder)

Alfredo Carella is the project supervisor and our contact person, if we have a problem we don't know how to solve. His objective is to help us to make progress and teach us the best way to succeed in our project.
He is a primary stakeholder because he can intervene at any time if the project goes in the wrong direction. We have to keep him informed of our progress.

- Evin Güler (OsloMet) (Secondary stakeholder)

Evin Güler is the supervisor of the Makerspace which is a working place that offers us all the tools we need. She has no interest in our project in particular but we have to inform her when we need to use the Makerspace.

- Teachers (Secondary stakeholder)

Schools and Teachers are the ones buying the MakeKit's kits to use them as tools during their lessons.
The instructions on the DIY kit have to be easy to understand not only for the students, but for the teachers as well who should be able to help the students with every occurring problem. So they are an interesting testing group, because the demand of the product depends on their willingness to use it as a teaching tool.

- Students (Secondary stakeholder)

Students are secondary stakeholders because they are the primary users of MakeKit's DIY kits. Students need safe and easy to use materials and simple instructions. While we're not necessarily informing them, we have to inform ourselves of their needs and competences to make sure they can use the toy kit and tailor instructions for them.

- Hobbyists (Secondary stakeholders)

Hobbyists are looking to learn by themselves or are perhaps interested in creating the product for fun. It's important for the hobbyists to be able to learn and create but also possibly make additions to the project as their own DIY ideas.

8.5. Battle:bit project description

Project description:

Project nr:	
Project name:	Sumo:bit / Car:bit / Vehicle:bit / Automo:bit / **Battle:bit**
Company:	MakeKit
Project Manager:	-
Kick-off	1.Feb
Deadline:	30. May
Launch	September

Project goal and strategy:

This project will be an intermediate level product.

Stimulate the imagination of children. Have competitions with other students about who has the fastest, strongest, agilest, car…

Requirements:

Must:

- The kit must differentiate itself from other competitive products.
- 3 major designs
- Pedagogic

Should:

- Use existing parts as much as possible
- Use recyclable material as much as possible
- Use the packaging as part of the design
- Be sturdy

U3- User, Usage, Usage:

- Users: 8 - 16 and up + their parents. (7 years with adult)
- Usage: A kit you build and code, it´s an add on kit for your micro:bit

- Usage: In the classroom and the users own home

USP:

- Freedom in design and modifications, but with “no brainer” examples to start with
- Competition in the classroom
- Repairable
- Remote controlled

Design principles:

Don´t use more material than necessary

Keep design simple and functional (Bauhaus, Dieter Rams)

Keep design friendly to the hand (reduce sharp corners)

Keep design playful and human-oriented

Keep design sturdy and avoid breakage

Main construction material: 3mm aircraft grade plywood.

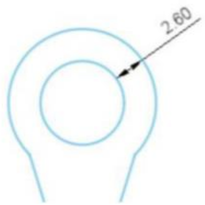
Secondary: 3mm MDF (only for parts that can not easily break)

Cardboard: Typically 3mm thick

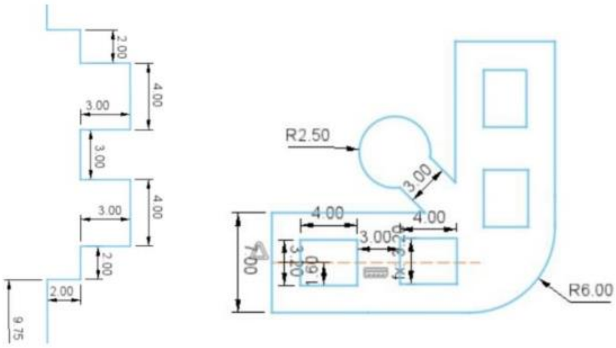
Screw: Nylon m3, different lengths, nylon nuts

Silicone o-rings, 7mm inner diameter x1.5mm thick, 28mm id x2.5mm thick

The real thickness of the plywood material is 3.2 and 3.5mm thick from factory

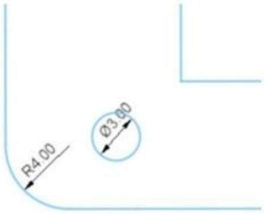


No part is to be thinner than 2mm, normally 3mm or more.



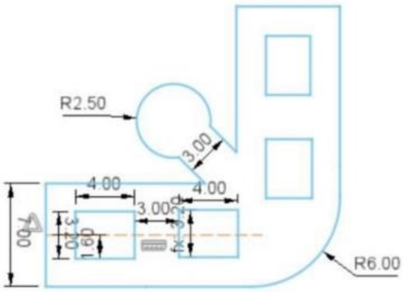
Connector: This is the MakeKit “system” connector that we would like to use further.

Male and female example from Hover:bit

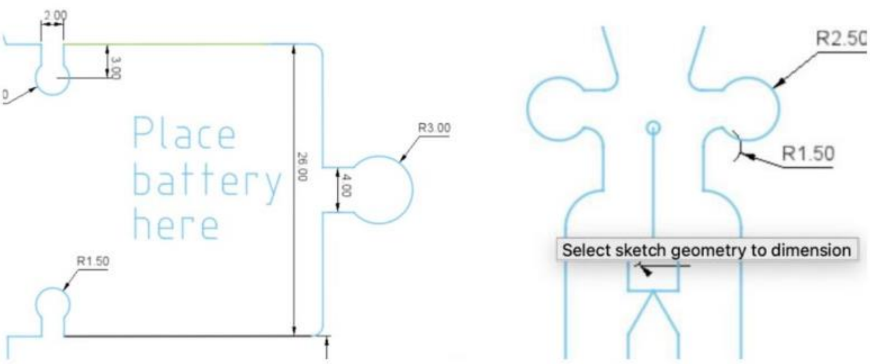


Rounded corners

We use rounded corners for outer corners on larger parts and typically in the outer edges of the assembled product. This stems back to the Micro:bit itself, which uses a radius of 3mm. Typical corners are rounded with 3-8mm, depending on the design look and feel.



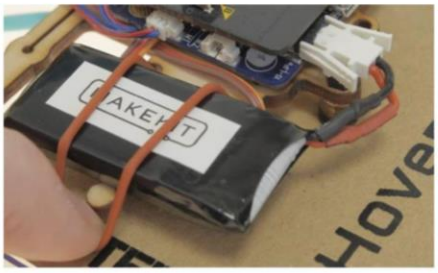
Corner joint from Hover:bit, larger corner is rounded, smaller corners not.

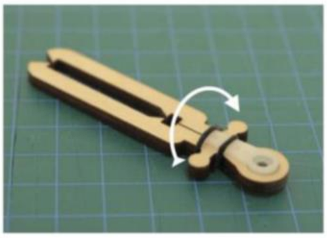


Rubber ring hooks

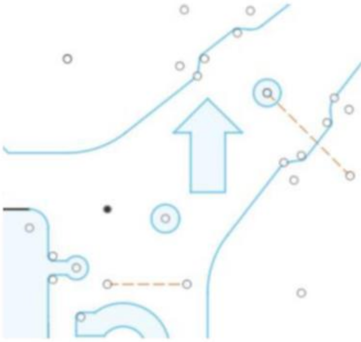
Inner and outer hooks for rubber rings 28mm diameter, and smaller knobs for 7mm rings. Knobs are 3mm or 2.5mm typically.

We use hooks for connecting rubber rings. Used for holding batteries, servos and more.

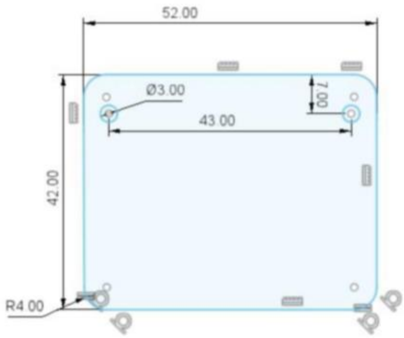


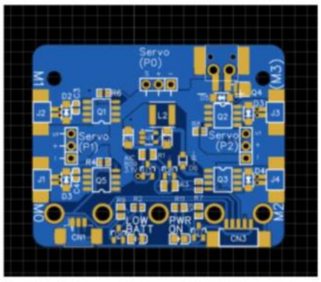


Breakage: If there is a risk of breakage, use rounded corners to distribute pressure.



On Air:bit, arms are rounded to avoid breakage in a crash.





Holder for multi:bit control board: Has the same footprint as the micro:bit, with screw holes for attaching the board. This design can be integrated into a base plate for the product. Control board seen on the right, the Micro:bit will be mounted directly on top.

Project cost:

Product development (materials etc):

First production:

Competitors:

Kitronix

4tronix

Makeblock

LEGO

Strawbees

Cannibalism of own products:

NO

Complexity and risk:

Fast rotating parts

Batteries

Fast car can cause damage

Working with tools

Resources:

All the maker tools of MakeKit

Our knowledge

Our network of companies and producers.

Video equipment

Offices etc.

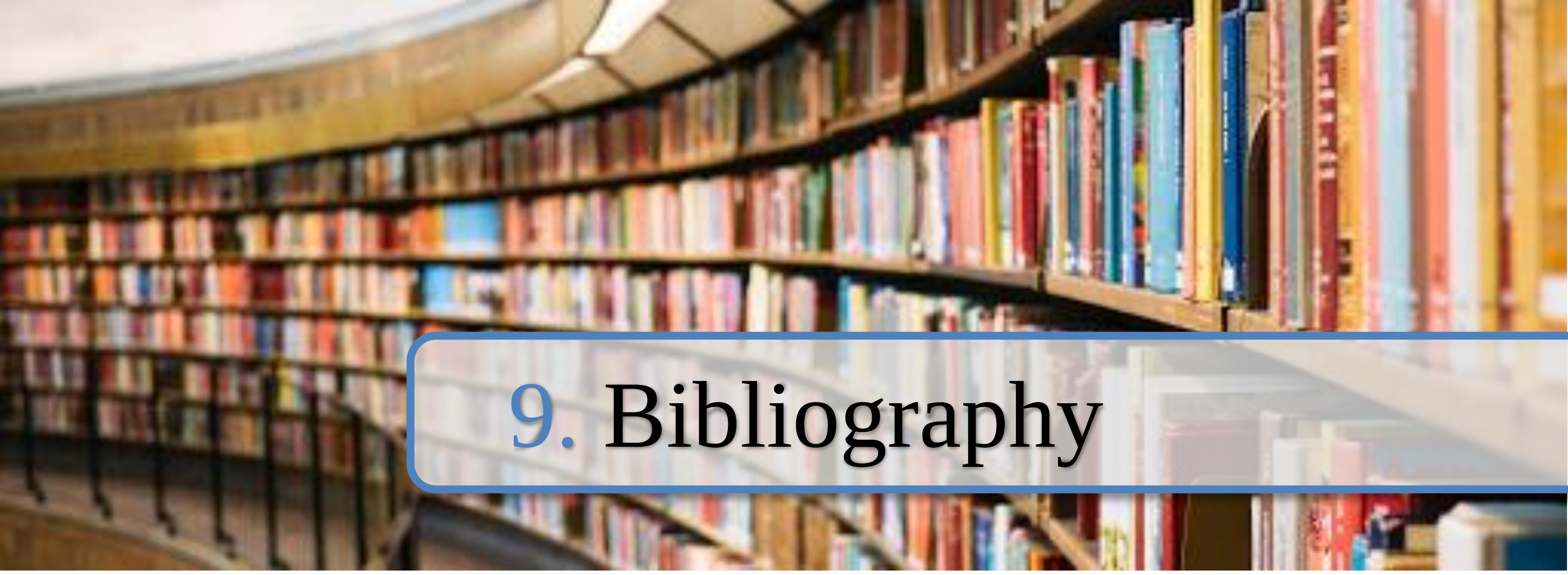
Links:

`https://www.feetechrc.com/`

`https://www.pololu.com/product/2818`

Key specs at 6 V: 0.10 sec/60°, 21 oz-in (1.5 kg-cm), 9.0 g.

https://www.feetechrc.com/continuous-rudder-machine.html



9. Bibliography

Materials:

- Cai, Z., Senalik, C. A., & Ross, R. J. (2021). *Wood Handbook. Wood as an Engineering Material*. [E-book]. https://www.fs.usda.gov/treesearch/pubs/62200

- Bajpai, P. (2018). *Biermann's Handbook of Pulp and Paper. Dans Biermann's Handbook of Pulp and Paper* (3e éd., Vol. 2, p. 177-185). Elsevier. https://doi.org/10.1016/C2017-0-00530-X

Laser cutting:

- Horvath, P. G. (2016). *Determination of the Minimum Intact Dimensions Available in Practical Applications of Laser Cutting*. Pro Ligno, 12, 9-16. http://www.proligno.ro/en/

- Jim, F. (2014). *Use a Laser Cutter/Engraver to Create Intricate Inlays*. (cover story). Tech Directions, 74, 14-17. http://www.techdirections.com/

- Eltawahni, H., Rossini, N., Dassisti, M., Alrashed, K., Aldaham, T., Benyounis, K., & Olabi, A. (2013). *Evaluation and optimization of laser cutting parameters for plywood materials. Optics and Lasers in Engineering*. 51(9), 1029-1043. https://doi.org/10.1016/j.optlaseng.2013.02.019

- EpilogLaser. *FUSION M2 32/40*. Laser System Manual (July 2019).

Robot & Tutorial:

- Cuartielles, D., Iriepa, N., Rodriguez, C., Lopez, E., & Garcia, J. (2019). *Educational Robots with Arduino: Annotated Prototypes.* Springer. `https://doi.org/10.1007/978-3-030-18141-3_13`

- Dibia, V. C., Ashoori, M., Cox, A., & Weisz, J. D. (2017a). *TJBot: An Open Source DIY Cardboard Robot for Programming Cognitive Systems.* [E-book] The Association for Computing Machinery, Inc. (ACM). `https://doi.org/10.1145/3027063.3052965`