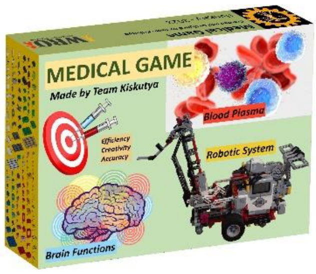
World Robot Olympiad – Future Innovators category (Senior) 2022

Team members: Hédi Zita KOVÁCS-BÁNHALMI, Sára MIHALIK, Lilla PETRÁNYI

Team name: „**Kiskutya**”

Hungary, Bányai Júlia Secondary Grammar School

Coach: Róbert KISS



MEDICAL GAME

Table of Contents

The Team

Our team consists of three members: Hédi Zita Kovács-Bánhalmi, Lilla Petrányi, Sára Mihalik, and we are team Kiskutya. The reason why we chose that for our team's name is thanks to our coach, who is also our informatics teacher. Whenever we need an example, he says „kiskutya.", so for all three of us, the first word/sentence we printed in python was „kiskutya", instead of the usual „Hello World!".

A little bit about our team:

All three of us are students at Bányai Júlia Secondary Grammar School in Kecskemét, and we finished grade 9 (Lilla and Hédi) and grade 10 (Sára) this year. We all started programming when we were in fifth grade (when we started studying in our current school), and the first programming language we learnt was EV3-G programming language. Now all three of us can program in Python and C++ as well.

Besides robotics and programming, we also like Mathematics, Physics and Chemistry, when we were young, we all enjoyed playing with LEGO, that's also one of the reasons why we got interested in WRO and programming.

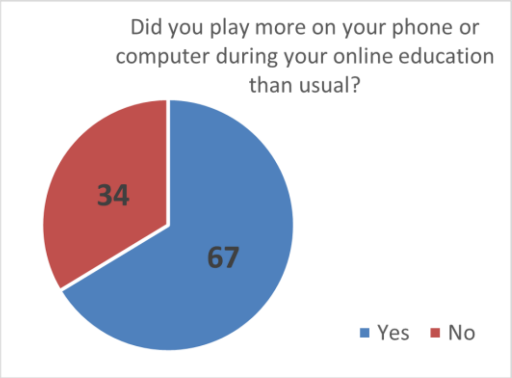
Even though we only started working together as a team recently, we can easily communicate and have fun together. Whenever we were working on our project, we always ended up listening to music (usually a Top Hits album on YouTube). There are two things we especially like doing after a tiring day of programming and working on our project, the first one is eating pizza together with our coach, and the other one is eating ice cream or cake at one of our local confectioneries.

Intruduction

Online games

The introduction of online education due to COVID 19 has significantly changed students' learning habits and activities in the online space. All this can be compared to a longer recovery period from an illness. Therefore, although we hope that the virus will not return, the following project may be valid even in social conditions that are partially isolated from the community. The development of social competencies and the fight against the fear of diseases and hospitals is one of the biggest challenges of today.

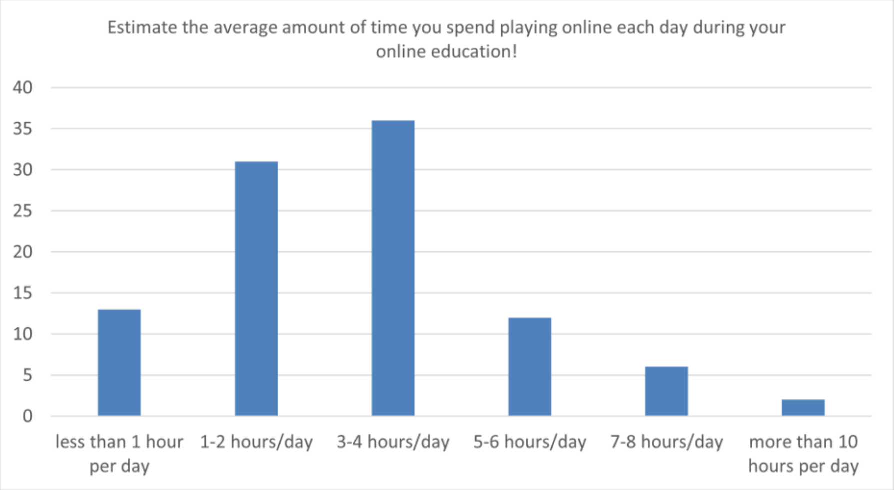
At our school, we conducted a questionnaire survey on activities during online education. The questionnaire was completed anonymously and 101 students answered the questions. The survey involved students aged 11-17. No significant difference in age breakdown was found, so the data were analyzed together.



In the questionnaire, we also asked about students ’time spent playing mobile phones.

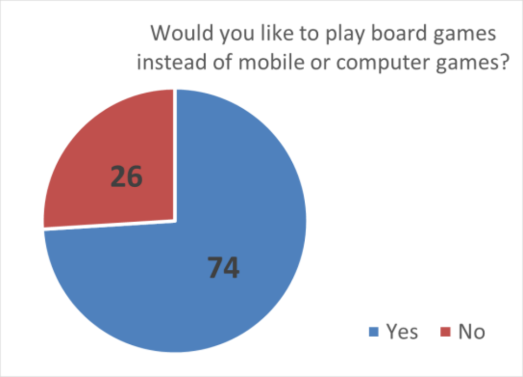
The response, as expected, showed that respondents played more computer games than before.

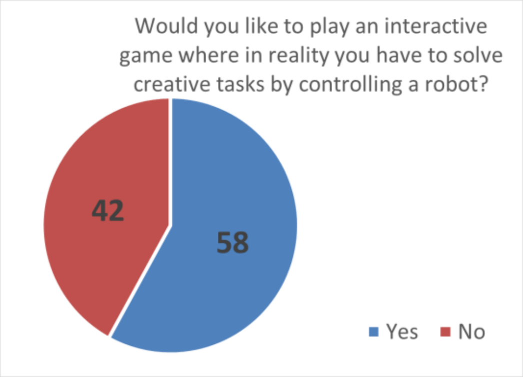
If we take a closer look at the data and are curious about the actual playing time on our mobile phones, the data we get is shocking.

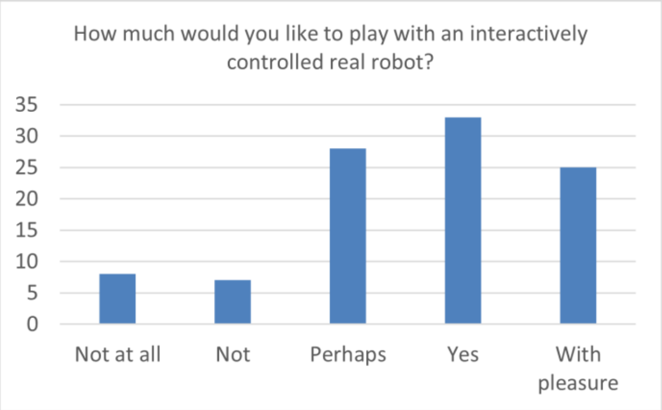


The average playing time per day was about 3.5 hours.

We were wondering what is the reason for this level of use of mobile phone games. We asked students if they would have liked to play board games or real robots.







The details of playing with real robots were not given, so each respondent decided for himself what he meant by "playing with real robots".

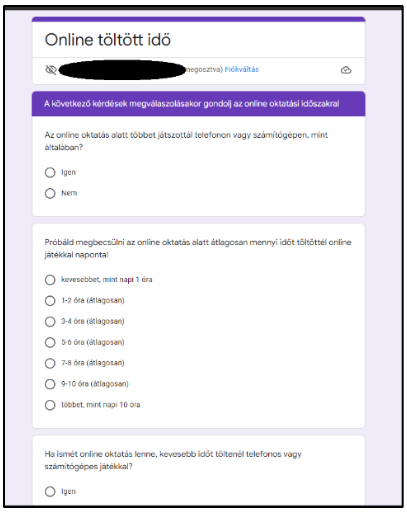
The latter issue has also been examined in more detail.

We concluded that students would be happy to play other types of games instead of mobile phone apps, but there was some barrier to this during online education. For example, there was no company to play with or there was no game (e.g., a robot) to play with.

The advancement of mobile games is transforming social competencies. It may not be a problem, but it seems to have effects that need to be corrected. [1]

Playing with video games improves motor skills, eye and hand coordination, but developing more cognitive skills requires more complex skills that can be effectively developed with activities other than video games. [2]

Our online survey



Online töltött idő

[redacted] megosztva) Fiókváltás

A következő kérdések megválaszolásakor gondolj az online oktatási időszakra!

Az online oktatás alatt többet játszottál telefonon vagy számítógépen, mint általában?

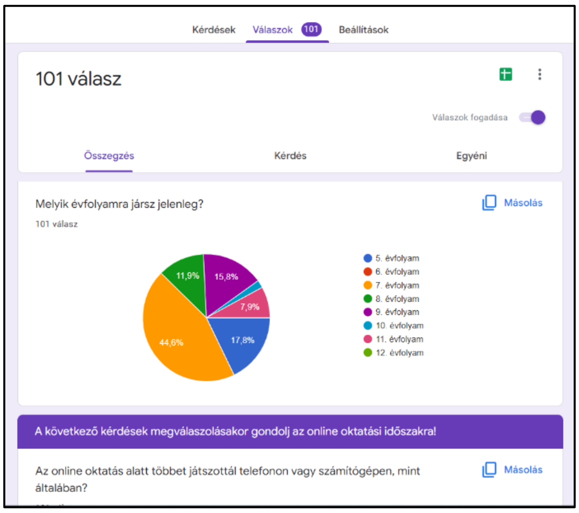
○ Igen
○ Nem

Próbáld megbecsülni az online oktatás alatt átlagosan mennyi időt töltöttél online játékkal naponta!

○ kevesebbet, mint napi 1 óra
○ 1-2 óra (átlagosan)
○ 3-4 óra (átlagosan)
○ 5-6 óra (átlagosan)
○ 7-8 óra (átlagosan)
○ 9-10 óra (átlagosan)
○ többet, mint napi 10 óra

Ha ismét online oktatás lenne, kevesebb időt töltenél telefonos vagy számítógépes játékkal?

○ Igen



101 válasz

Válaszok fogadása

Melyik évfolyamra jársz jelenleg?

101 válasz

A következő kérdések megválaszolásakor gondolj az online oktatási időszakra!

Az online oktatás alatt többet játszottál telefonon vagy számítógépen, mint általában?

Fear of the hospital and the doctor

Half of children are afraid of going to the doctor, according to a new survey of 726 parents. And some kids get so upset that 1 in 5 parents say they find it hard to concentrate on what the doctor or nurse is saying. Some also say they had postponed a vaccination or cancelled an appointment due to their child's fear.
Result revealed that more than 90% of children said they were afraid of at least one thing in a hospital. [3] [4]

One of the most common ways of getting rid of your fears is to face it, however that can be quite overwhelming for such a young child.

Children with needle phobias try to avoid any contact with needles and for them the thought of coming into contact with the cause of the phobia causes anxiety and panic.

Based on the experience of the last two years, all this has become particularly important. (Example: COVID 19 - Vaccination)

Conclusions

Based on the above introduction, we have formulated some ideas that were the starting point for our project.

- The game is a really important part of kids' skill development.
- Games based on digital technology are unavoidable.
- Devices that go beyond computer games are needed to develop the ideal competencies of children.
- It is necessary to overcome children's fear of medicine.
- Children need to be introduced to the achievements and need for medicine, all in a playful way
- In addition to overcoming fear, spreading knowledge is also important.
- The development of logical thinking, problem solving and programming competencies is an important element of the 21st century.

Reflections

- We will create a game in which a robot can be used to solve problems related to medicine.
- The robot can move remotely and automatically. It has arms that allow you to perform complex movements.
- The problems to be solved are related to medicine and psychology.
- Everything takes place in a real, built environment.
- On the playing field, the robot goes automatically to each task, but when it gets there, it solves the task remotely. The developed computer program measures and stores the time required for the player to complete. In response to the rankins, a single-player competition is held.
- The problems that need to be solved can be modified and changed, since they are module-like.
- Even a commercial available game can be built around them.

THE MEDICAL CONCEPT OF OUR PROJECT

The background of the 3 health problems in the project and the game.

Blood plasma

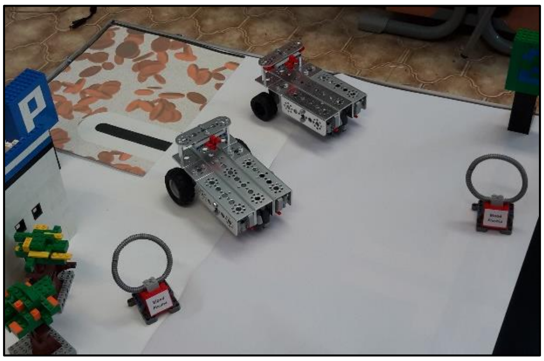
Blood plasma is a pale, amber-colored liquid that is 90 percent water and 10 percent protein. It is a vital component of blood formation and cannot be created artificially.

Similar to blood donation, plasma donation involves merely collecting blood plasma while the cellular components are returned to the body along with saline. Life-saving medications are made from the extracted protein in need to treat disorders including immune deficiency, liver failure, burns and bleeding.

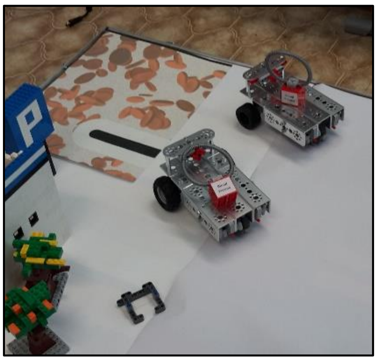
Anyone who is a healthy adult between the ages of 18 and 60, weighs between 50 and 140 kg, and passes an eligibility test can donate blood.

All plasma donations begin with a data match and a medical examination. The collected blood plasma is collected in a bottle and the cellular components are returned to your circulation along with saline to replenish the lost fluid. The entire procedure takes a total of one hour. [5]

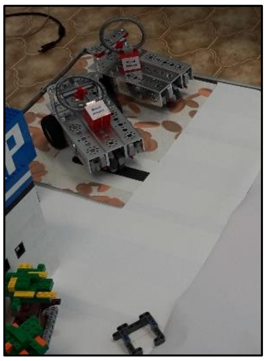
In the game, the task is to place the LEGO element, which symbolizes the blood plasma, on the transport trolley and move the two trolleys into the hospital storage room with the help of the robot.



Starting position



Plasma ont he trolley



Task completed

Left or right brain dominance

The brain consists of two hemispheres, each of which performs a number of tasks. The two hemispheres of the brain communicate with each other. The left hemisphere controls the muscles on the right side of the body, while the right hemisphere does the opposite. For this reason, damage to the left hemisphere of the brain, for example, can affect the right side of the body.

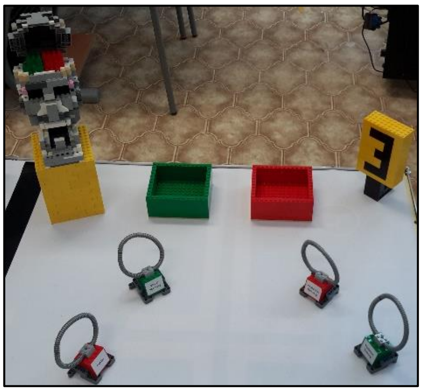
According to the theory of left or right brain dominance, each brain hemisphere controls different types of thinking. It is also claimed that people prefer one way of thinking to another.

Some of the skills associated with the right brain are recognizing faces, expressing and reading emotions, using imagination and creativity, intuitiveness, and color perception.

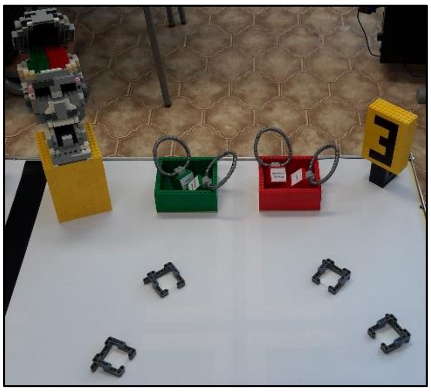
The left brain is considered expert at tasks involving logic, language and analytical thinking. The left hemisphere is described as better at language, logical and critical thinking, counting, and reasoning.

In subjects such as mathematics, both hemispheres of the brain work most strongly together. [6] [7] [8]

In the game, the task is to pair colored LEGO elements that symbolize different human abilities with the corresponding hemispheres of the brain. The batteries must be placed in the appropriate container. Elements that symbolize human capabilities can be changed.



Starting position

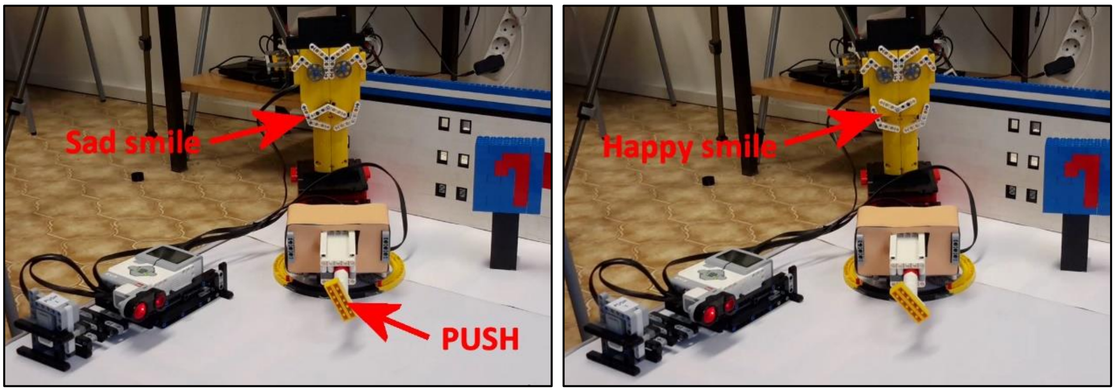


Task completed

Vaccination

In our project one of the games is designed to help children get less scared of getting a shot by giving a vaccination themselves.

Children with needle phobias try to avoid any contact with needles and for them the thought of coming into contact with the cause of the phobia causes anxiety and panic. That’s why our team thinks that we should provide as many steps as possible, that including the part of this game. In the game they have to push a needle built from LEGO into a block symbolizing the human skin. Since the needle is not real, children won’t be afraid of it and by playing the game they will also understand that vaccines are harmless and why they are so important. To make the game more difficult and enjoyable, the block also moves that way the player will have to find the right angle. Upon successful completion, the originally sad face will smile.



Starting position Task completed

Game Description

In our project, we created a game similar to video games. Our game has to be played in real space with a remote-controlled robot.

There are currently three tasks in the game in an area of 150 cm x 160 cm.

The remote rover starts from the base and automatically moves to the selected task. When you reach the target area, you can switch to remote control mode. As soon as the robot enters the target area, the time measurement starts automatically, which you can keep track of on a laptop screen. The three tasks are previously described vaccinations, blood plasma and hemispheric functions.

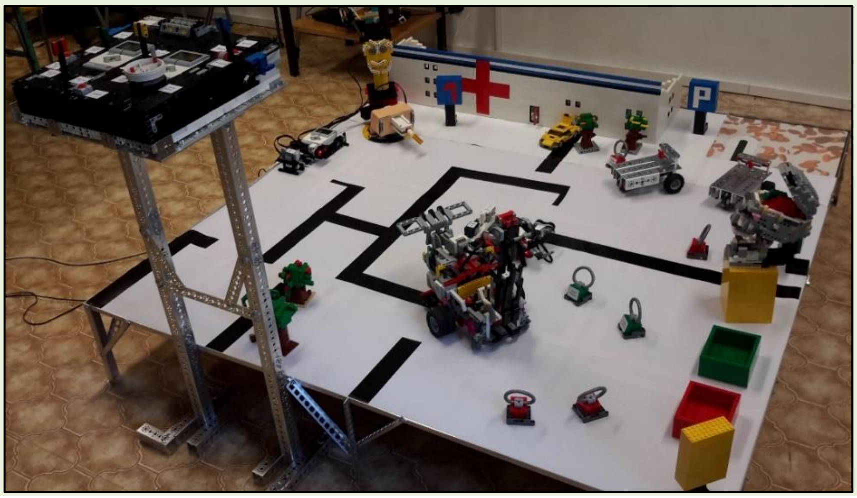
Once the player has solved a task, a push button must be pressed with the rover. The timer then stops and the players position is displayed on the laptop screen. The results are continuously stored in a file.

The following step is to utilize the remote control to choose the task's serial number, after which the rover will proceed there on its own. You can even solve the same task several times in a row.

The philosophy of the game is the same as the principle of well-known mobile phone games. However, here the tasks need to be completed in a real (3D) environment.

The game is not easy. If we take into account the results obtained in our questionnaire survey, a professional level can be achieved with 3.5 hours of practice per day. 😊

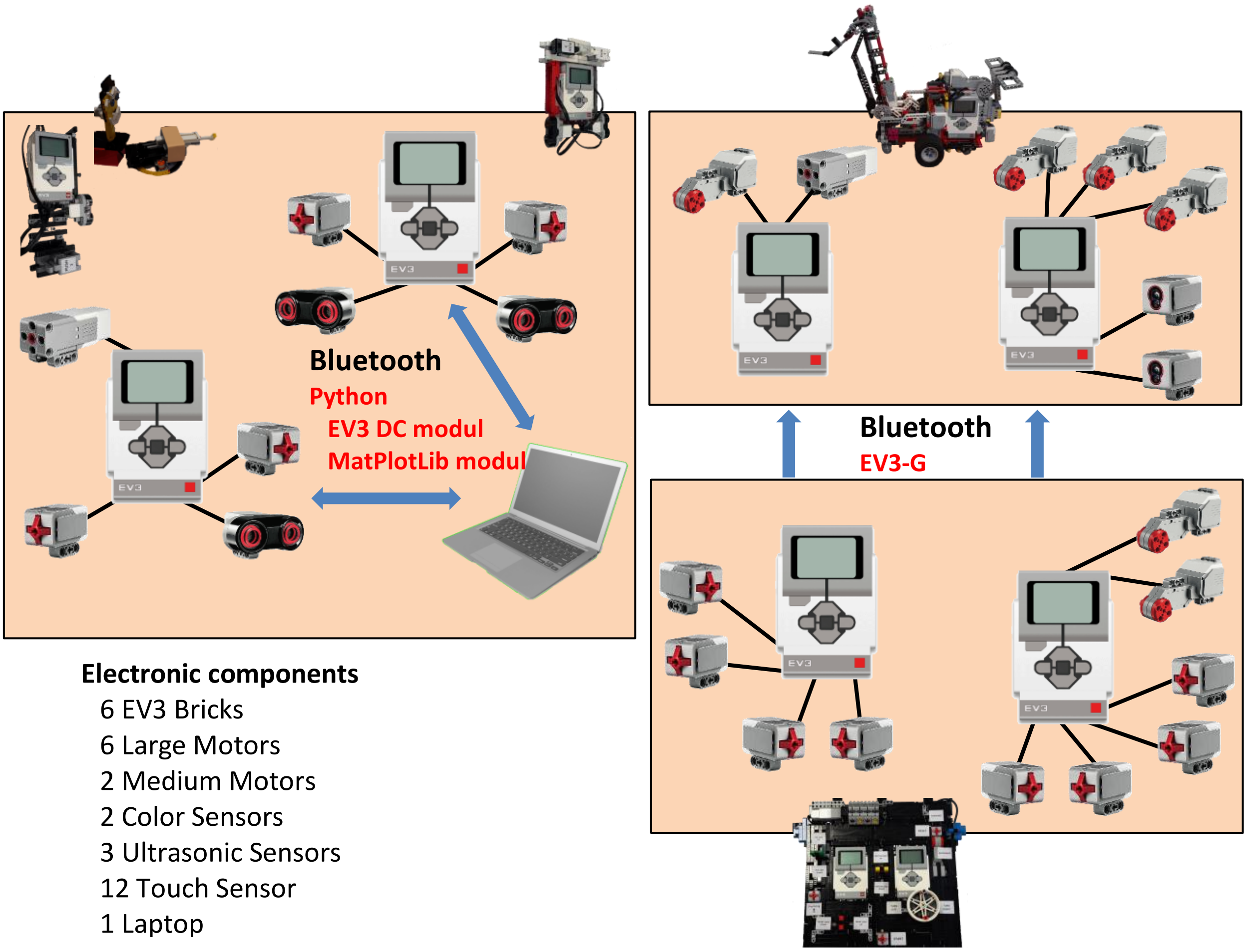
At the same time, it develops spatial vision, complex movement coordination, dexterity and logical skills much more complexly than mobile phone games. At the conclusion of the report, we will discuss this topic in greater detail.



The Robot and Its Function

Network structure and communication between robots

We used 6 EV3 bricks in the project.
There are 2 bricks in the remote control robot (rover) and 2 in the remote control device.
One brick in the rover is responsible for moving the robot, while the other brick moves the manipulator arms.
One of the bricks of the remote control device sends the control signals for movement and the control mode can also be set here. There are two control modes to choose from: automatic movement along the lines connecting the tasks, and manual remote movement when solving tasks. The other brick of the remote control device sends control signals to the manipulator arms.
Data is sent via bluetooth. Two bricks are connected to each other.
We used two bricks to detect when a player started solving one of the tasks. These bricks are also responsible for measuring the playing time. Here we have chosen a new way of communication. The two bricks are connected to a laptop via bluetooth. The code was written in Python. We used the Python EV3 DC add-on module. In this case, the bricks receive control signals from the laptop's Python program (Direct Control). There is no program running on the bricks, we only use the sensors via the laptop. Thus, both the measured values and the calculated data are generated on the laptop. We can create a graphical representation of this data with a Python add-on module (matplotlib).
The following figure shows the network structure and the connections of the sensors used.



Electronic components

- 6 EV3 Bricks
- 6 Large Motors
- 2 Medium Motors
- 2 Color Sensors
- 3 Ultrasonic Sensors
- 12 Touch Sensor
- 1 Laptop

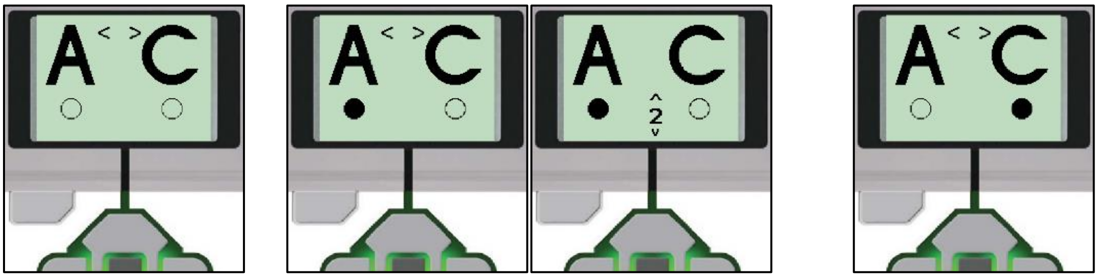
The rover and the remote control

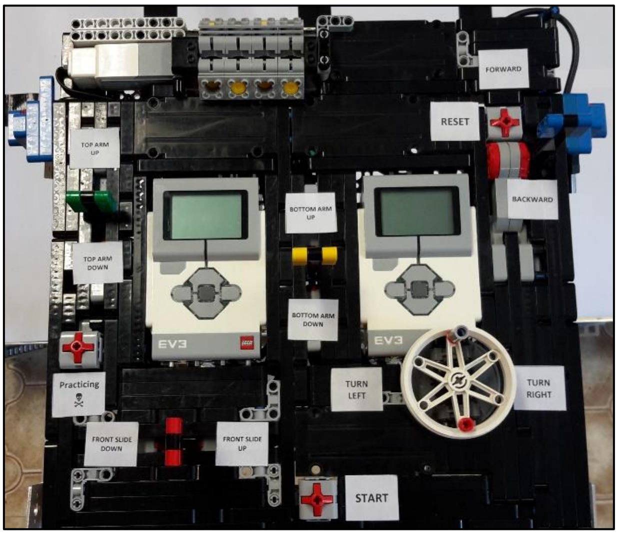
Mechanics and programming (EV3-G)

We built a lot of constructs in the project and the program code is quite long. Only a few typical details are highlighted below. Our code written in EV3-G contains a total of 248 blocks organized into 13 My Blocks. Each of the 4 bricks has a main program that operates the system by calling its My Blocks.

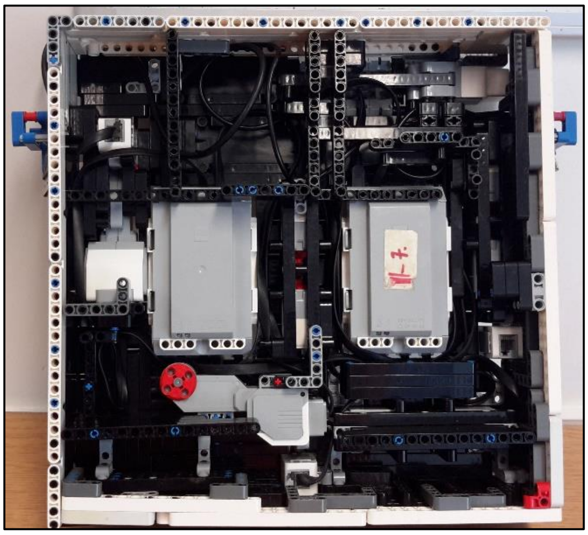
After building a bluetooth-based network, the right brick on the remote control console will launch the program.

The first step is to select the operating mode. It is possible to use the system in Remote mode when the player solves one of the three tasks, or it can be used in Automatic mode when the robot moves independently on a fixed path between two tasks. In automatic mode, the destination area number must also be selected.



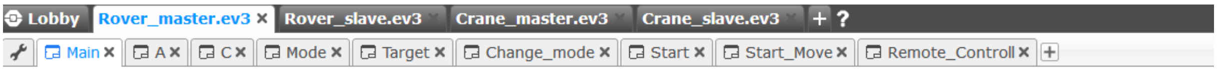


Remote control (front view)

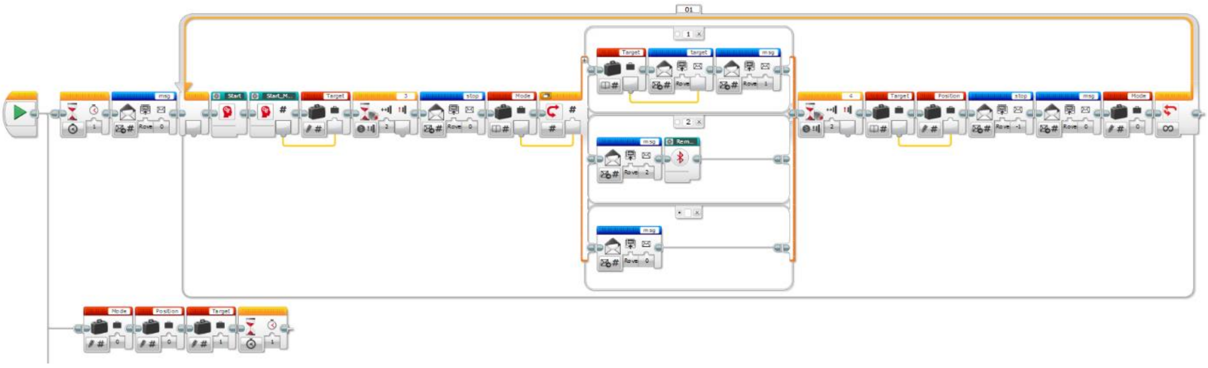


Remote control (bottom view)

The Remote Control Consol Programs

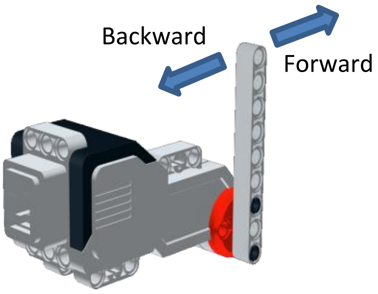


The graphic below depicts the code snippet for the main program of the remote control console.

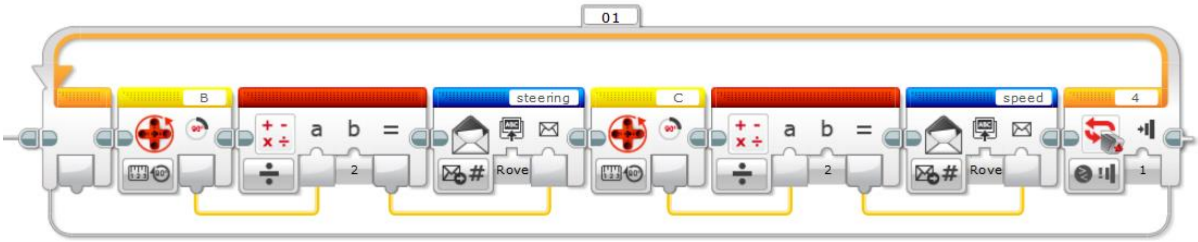


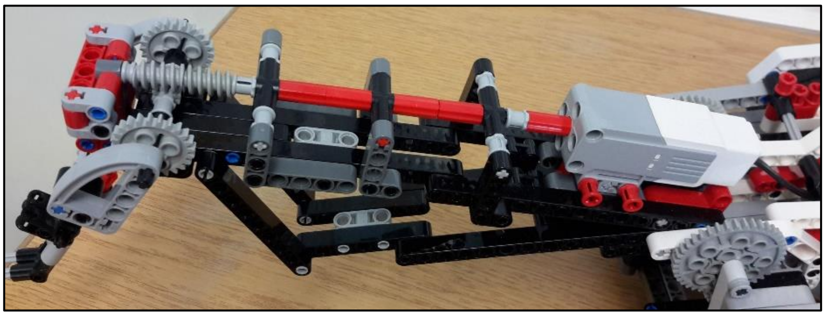
The *Mode* variable stores the selected game mode, *Position* **is** the current position of the robot, and *Target* is the target area to which you need to move in automatic mode. Depending on the contents of the *Mode* variable, the instructions on the different threads of the Switch will run. The values of the variables are transmitted to the rover motion control brick via Bluetooth.

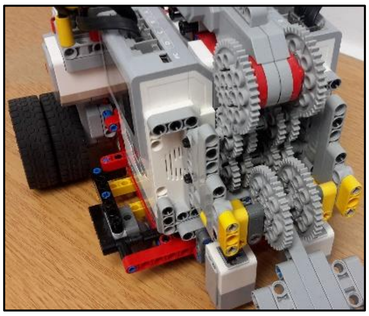
In remote control mode, the sign of the rotation of a motor-mounted arm is sent to the rover by the corresponding My Block. The rover interprets the values obtained as the engine speed. For example, the motor will run at a speed of -90 if the lever is moved by -90 degrees. The steering was solved with the help of another engine with a similar principle.



The My Block code of the sender

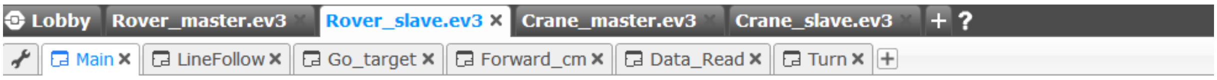






The brick responsible for the rover's movement receives messages from the remote control console. The main program chooses between two options, depending on the mode of operation the player has chosen.
If the player uses the rover in remote control mode, the two motors rotating the wheels will receive the transmitted speed and steering values. It will move in line with that.

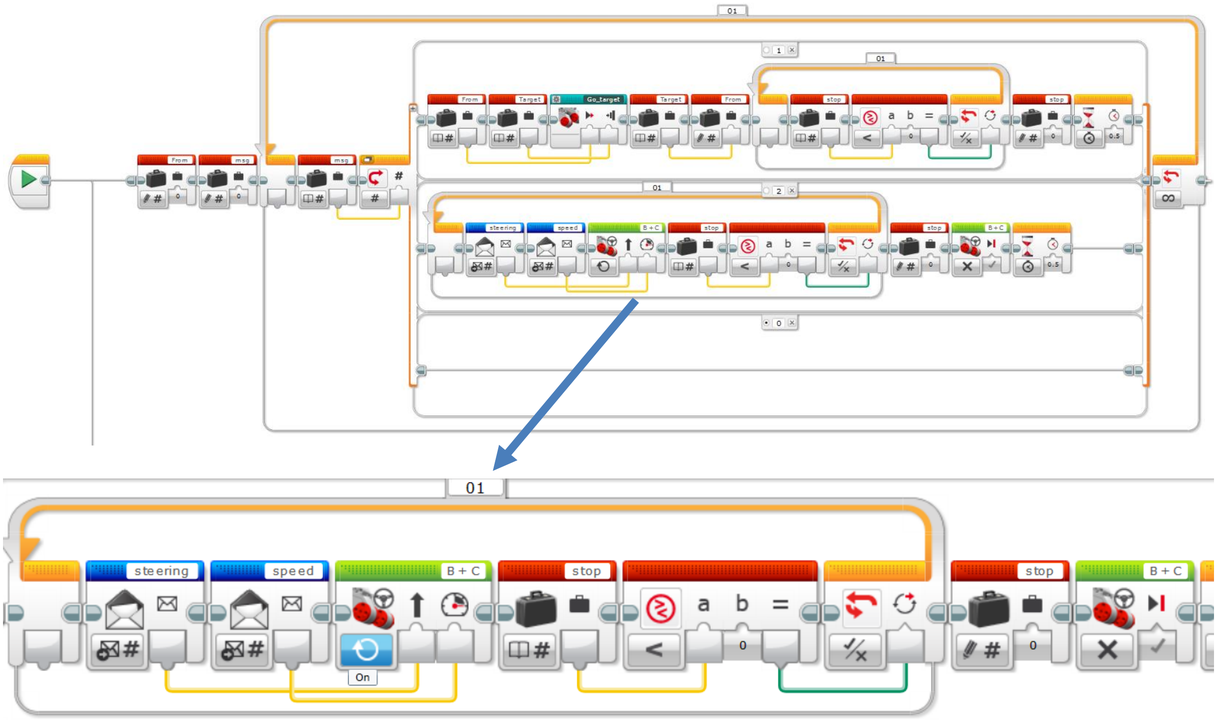
The Rover programs



Lobby | Rover_master.ev3 | Rover_slave.ev3 × | Crane_master.ev3 | Crane_slave.ev3 | + ?

Main × | LineFollow × | Go_target × | Forward_cm × | Data_Read × | Turn × | +

The Main program



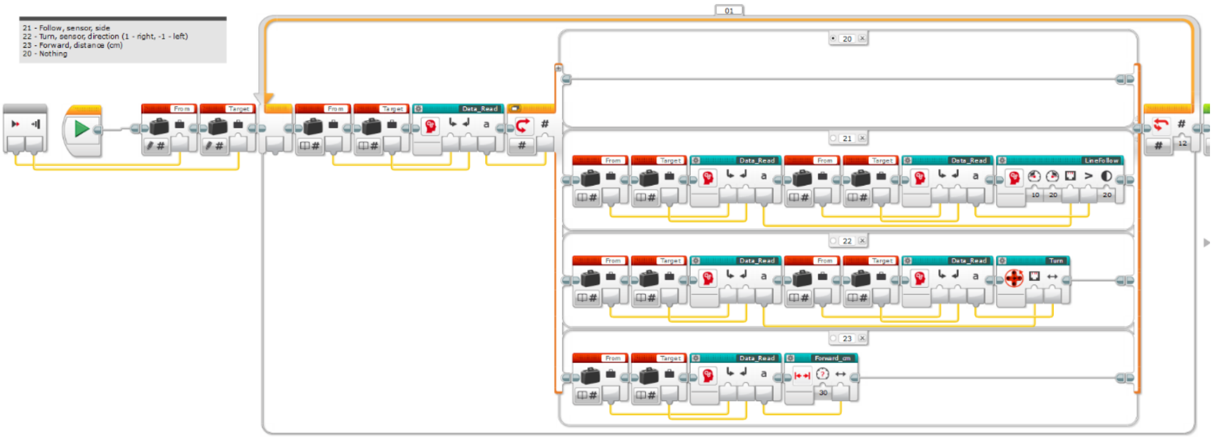
Once a task is completed the player has the option of continuing with the robot to the next one or starting over. On the home screen, you must select the automatic mode on the remote control console. Here you can specify which target area the robot should move to. The current position is known.
Because the route is different between each target area, but we wanted to handle it uniformly in the program, we created a virtual track description.

The robot has two color sensors. As you move, the black path will be between the two sensors. Each route can be described by a combination of three modes of movement.

1) The robot follows the edge of the path. You must specify with which sensor and which side of the route to follow. Tracking lasts until the other sensor detects black.

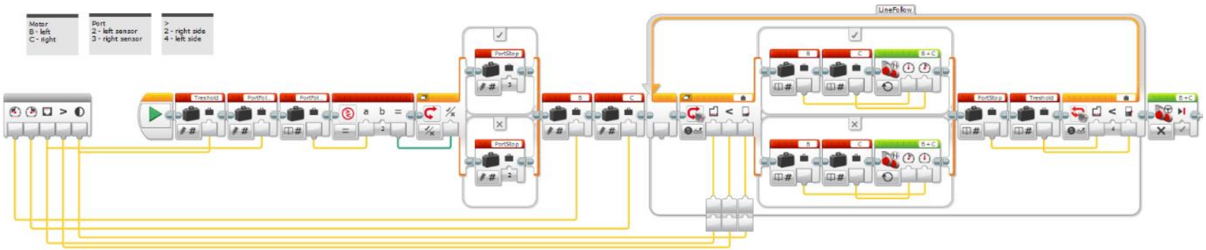
2) The robot turns upon reaching an intersection. You must specify in which direction and with which sensor to monitor the end of the turn. (As long as the sensor detects black.)
3) The robot goes straight ahead for all distances (centimeters). Progress is needed to pass over the intersection.

Because paths between tasks consist of a different number of motion forms, a fourth control value is required for which nothing happens.
The motion elements needed to traverse the routes are stored in files. The file name is a two-digit number. The first digit is the starting position, the second digit is the target position. When the program is running, the My Block receives the start and destination ID numbers and opens the file with that name. For example, to move from area 1 to area 3, the target file is in the 13.rtf. The longest path consists of 12 motion elements, so each file is supplemented with empty instructions so that it contains 12 elements.



21 - Follow, sensor, side
22 - Turn, sensor, direction (1 - right, -1 - left)
23 - Forward, distance (cm)
20 - Nothing

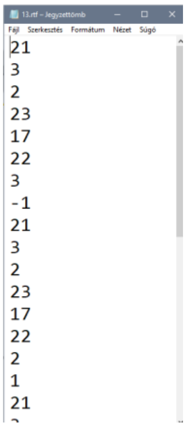
For example, the line follower block, turn block and a detail of 13.rtf.



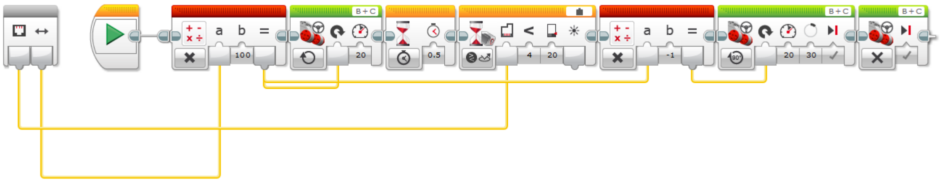
Motor
B - left
C - right

Port
2 - left sensor
3 - right sensor

>
2 - right side
4 - left side

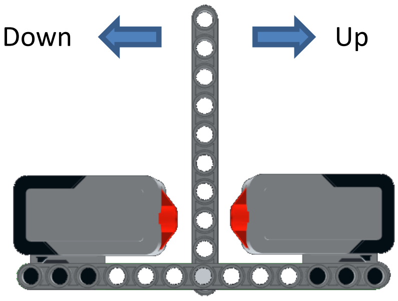


```
21
3
2
23
17
22
3
-1
21
3
2
23
17
22
2
1
21
```



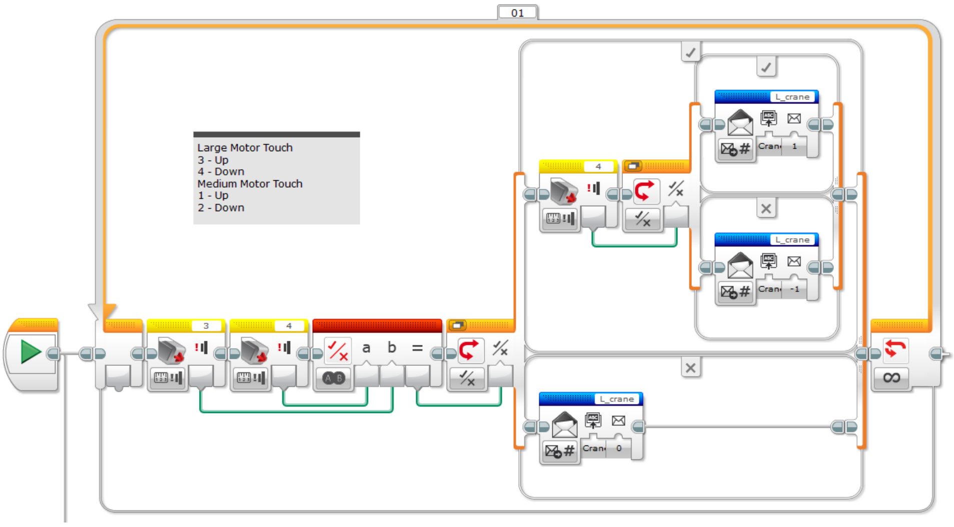
With this method, we were able to handle any route traversal with uniformity. We created a total of 12 files with the names 01.rtf ... 33.rtf.

The Crane and the manipulator arms



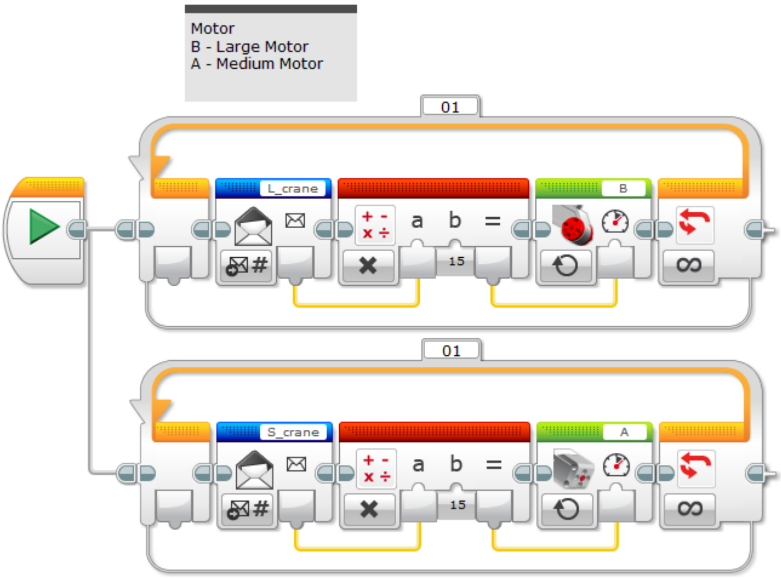
The other brick in the remote control console is responsible for controlling the crane and the manipulator arms.
Motion control here is based on other mechanical and programming principles.
There is a touch sensor on each side of a vertical arm. By moving the lever, any impact sensor can be pressed, but not more than one at a time. The two touch sensors are responsible for moving up and down.

The main program of the remote control console sends the current status of the touch sensors via bluetooth.



Large Motor Touch
3 - Up
4 - Down
Medium Motor Touch
1 - Up
2 - Down

The appropriate brick of the rover will move the crane arms down or up based on the data received.



Motor
B - Large Motor
A - Medium Motor

THE GAME MODE

An important element in the game is measuring the time it takes to complete each task. The rankings and top lists in video games are based on this, thus all of this is significant there as well. To calculate playing time two problems have to be solved. For instance, we intended to use a dynamically updating chart to display the passing of time and the leaderboard at the end of the game. The Python programming environment seemed ideal for this. Because the brick screen is too small, the graphs appear on a laptop screen. The other issue was how to signal to the laptop's software that the player had begun and completed the assignment. The measurement data will be saved in the brick's memory if you execute written code on it, and the laptop won't be able to evaluate the data.
Therefore, we chose a Python module developed for an EV3 robot, where the code runs on the laptop, the data is also stored. The module is called EV3 DC (Direct Control). [10] The laptop sends the control signals to the brick via a bluetoot connection, which it executes. There is no need for any program to run on the brick, it is enough to be on. The firmware of the brick can be any. With this module, the brick sensors can be accessed directly from the laptop, so we can read measurement data and store them in files and lists on the laptop.
There are several different modules available in Python for graphical display. We chose the matplotlib class library developed specifically for mathematical statistics. [9]

Measuring the completion time of tasks

EV3 DC – Python

First, we imported the program's necessary structures and classes. The bricks and sensors were then determined. The connection is based on the brick's bluetooth card ID.

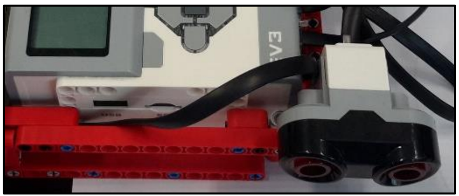
```
import ev3_dc as ev3

brick1 = ev3.EV3(protocol=ev3.BLUETOOTH, host='00:16:53:54:6f:83')
brick2 = ev3.EV3(protocol=ev3.BLUETOOTH, host='00:16:53:61:9c:15')
```

We created instances of the sensors from the DC classes. We can achieve their current values without running a program on the brick. And we can control the motors.

```
touch11 = ev3.Touch(port=ev3.PORT_1, ev3_obj=brick1) #task 1, stop touch
touch13 = ev3.Touch(port=ev3.PORT_3, ev3_obj=brick1) #task 1, Syringe touch
ultra11 = ev3.Ultrasonic(port=ev3.PORT_2, ev3_obj = brick1) #task 1, start ultra
motor1b = ev3.Motor(ev3.PORT_B, ev3_obj=brick1)  #Smile-sad
motor1a = ev3.Motor(ev3.PORT_A, ev3_obj=brick1)  #Syringe, randomize
touch22 = ev3.Touch(port=ev3.PORT_1, ev3_obj=brick2) #task 2, stop touch
touch21 = ev3.Touch(port=ev3.PORT_3, ev3_obj=brick2) #task 3, stop touch
ultra21 = ev3.Ultrasonic(port=ev3.PORT_2, ev3_obj = brick2) #task 2, start ultra
ultra22 = ev3.Ultrasonic(port=ev3.PORT_4, ev3_obj = brick2) #task 3, start ultra
```

The main part, in our program is the time measuring function. The measuring starts when the ultrasonic sensor detects the robot.

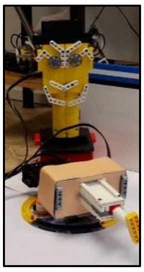
```
if (ultra21.distance < 2.0): #task 2
        start = time.time()
        winsound.Beep(440, 500)
        deltaido = idotmer(touch21,start)
```

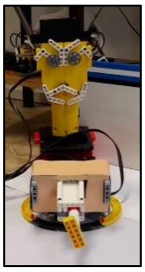


The *idotmer()* function provides time measurement and graphical display. The operation of this function will be discussed in the next section.

In Task 1 (Viruses and Vaccines), the timing is supplemented by a random number draw. When the robot enters the play area, the system randomly rotates the syringe and makes the face sad. Thus, the robot can only solve the task after finding the right position.

```
start = time.time()
        fok = random.randrange(100,180)
        motor1a.start_move_to(fok, brake=True)
        motor1b.start_move_to(180, brake=True)
        winsound.Beep(440, 500)
        deltaido = idotmer(touch11,start)
```



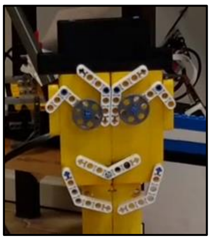


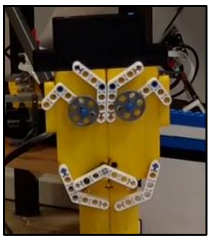
If the syringe is pushed in with the robot, the face will smile.

```
        if (touch13.touched):
            motor1b.start_move_to(0, brake=True)
```

After completing the task, the syringe will return to its home position as well as the smiling face.

```
        motor1a.start_move_to(0, brake=True)
        sleep(1)
        motor1b.start_move_to(90, brake=True)
        sleep(1)
        motor1b.start_move_to(0, brake=True)
        sleep(1)
```

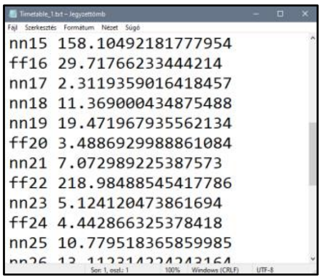




The completion of the task can be signaled to the system by the player pressing a button labeled PUSH with the robot. The timer will stop and the system will generate a username, which will be written to a file along with the measured time.

The file is read back into a list and sorted by time.

```
    adatok.append([nev,deltaido])
    ki = open("Timetable_"+str(tazon)+".txt","a")
    ki.write(nev+" "+str(deltaido)+"\n")
    ki.close()
    adatok.sort(key=rendez)
```


Timetable_1.txt – Jegyzettömb
Fájl Szerkesztés Formátum Nézet Súgó
Sor: 1, oszl.: 1
100%
Windows (CRLF)
UTF-8

```
nn15 158.10492181777954
ff16 29.71766233444214
nn17 2.3119359016418457
nn18 11.369000434875488
nn19 19.471967935562134
ff20 3.4886929988861084
nn21 7.072989225387573
ff22 218.98488545417786
nn23 5.124120473861694
ff24 4.442866325378418
nn25 10.779518365859985
nn26 13.112314224243164
```

After completing the task, the system asks if the completion was successful. The time ranking is then displayed.

In the case of Task 1, performance can also be automated, as we have built-in a sensor for sad-smile face display. The other two tasks (Blood Plasma, Hemisphere Functions) have no sensory monitoring. As an improvement on the game, this can be expanded.

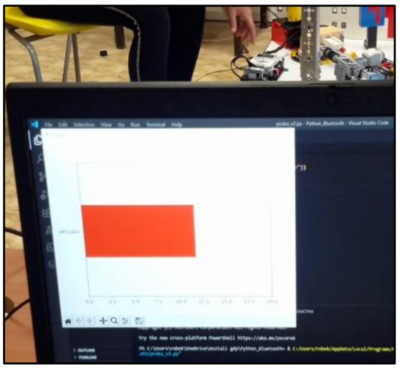
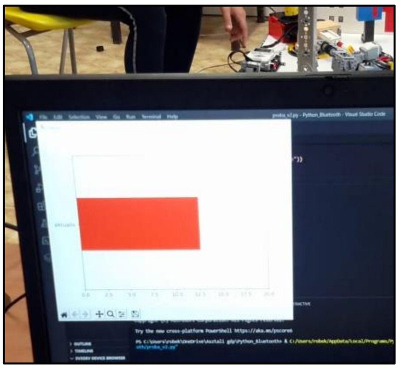
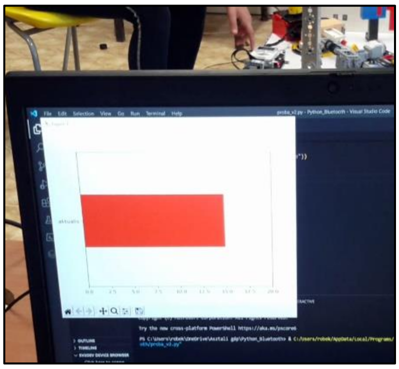
The graphic display – Python

Matplotlib

The matplotlib class library is an add-on to Python that can display charts in a graphical window.

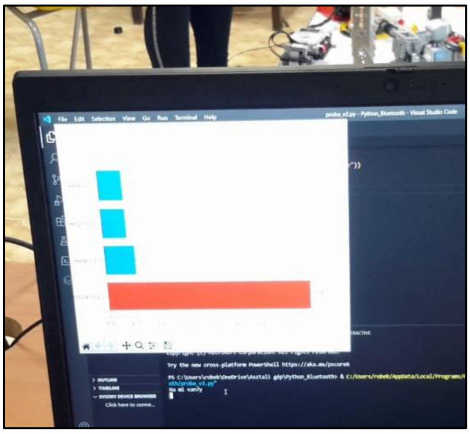
```
import matplotlib.pyplot as plt
import numpy as np
```

When the player starts solving a task, the timer starts and the system displays the elapsed time on the laptop screen in the form of a dynamically changing chart.

```
plt.show()
    axes = plt.gca()
    axes.set_xlim(0, maxx)
    axes.set_ylim(-1, +1)
    while not azont.touched:
        ido = time.time()
        y = [ido-start]
        plt.barh("aktualis",ido-start,color=("red"))
        plt.draw()
        plt.pause(1e-17)
```

When the task is completed, the player's time as well as the first 3 positions in the time ranking will be displayed on the screen. The player's graph is displayed in red. After placing a player, his position is shown in parentheses. The data is stored in a file and is not deleted, so the recorded data is retained "forever".



```
x = np.array([nev+" ("+str(hely)+".)",adatok[dobogo[2]][0]+"
(3.)",adatok[dobogo[1]][0]+" (2.)",adatok[dobogo[0]][0]+" (1.)"])
    y = np.array([round(deltaido,
2),round(adatok[dobogo[2]][1],2),round(adatok[dobogo[1]][1],2),round(adatok[dobogo[0
]][1],2)])
    plt.cla()
    for index, value in enumerate(y):
        plt.text(value, index, str(value))
    plt.barh(x, y, color=("red","blue","blue","blue"))
    plt.show()
```

Conclusions and plans

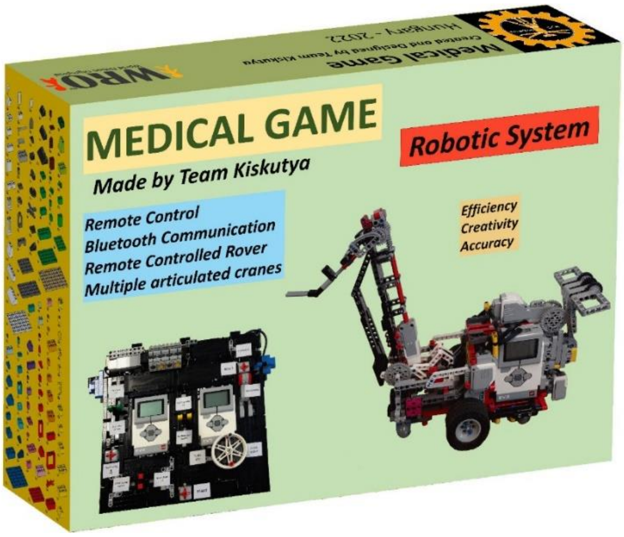
The idea of a commercial game

During the development of the game, our team remembered that the idea could be turned into a commercially available game.

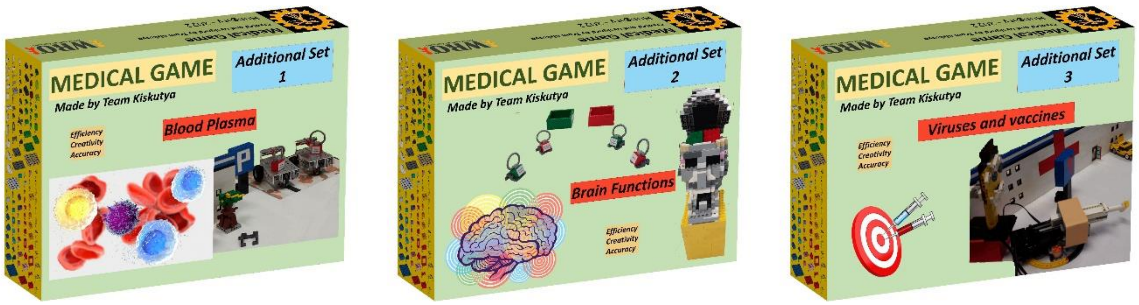
According to our concept, a multi-level game could be developed on the model of video games, where the tasks have to be solved with a robot moving in a real environment.

The topic of the tasks can be anything, not just medical.

The marketed game would be modular. The basic kit would include a remote-controlled robot and a controller. This could be used on its own. Constructions and paths for each task could be obtained in the form of additional sets. Basic electronics are not necessarily LEGO. We also only partially used the programming capabilities of the EV3 bricks, as the code of the timing and evaluation system ran on the laptop, we only used the sensor for the bricks.



The Base set

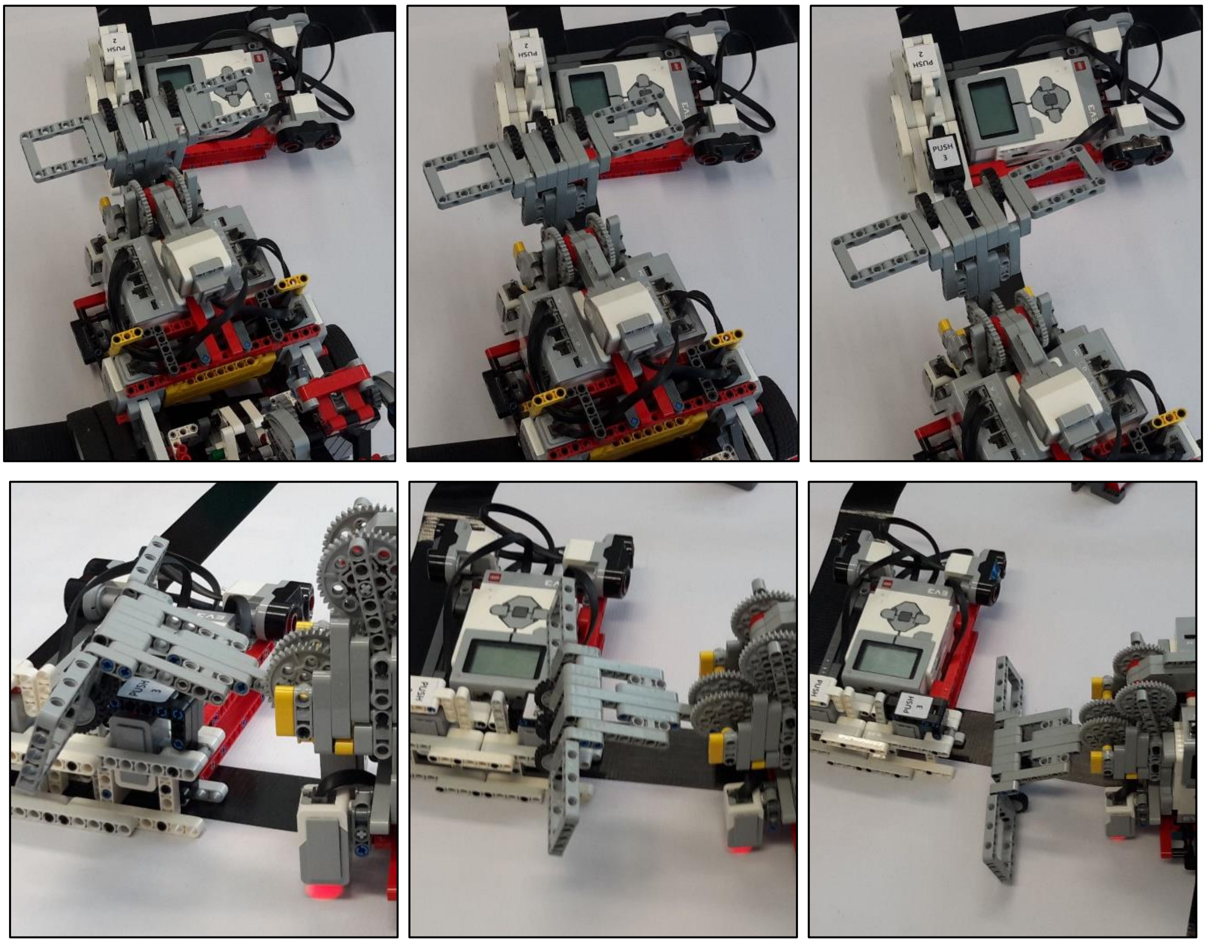


Additional sets

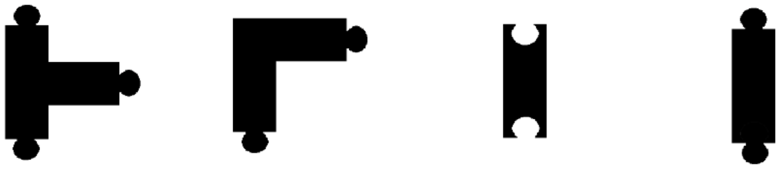
The game on the market is basically based on dexterity and practice. (Like most video games.) That’s why it can be good fun for a wider age group.

Adding sets for tasks from different disciplines increases the variety. By incorporating science, the educational effect can be enhanced.

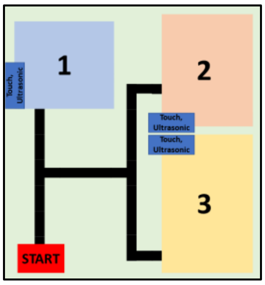
It improves spatial vision and motion coordination much better than video games. For example, the following three images show the position of the robot from the player’s perspective. In our project, at the end of the task, a button labeled PUSH must be pressed with the robot arm. There is little difference between the three images, although only in the middle image is the robot in a good position.

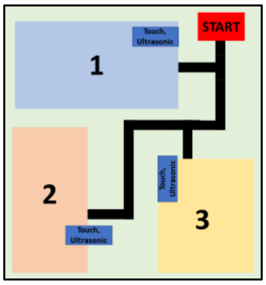


The variety of the game can be further increased if the path between the individual tasks is not fixed, but can be built freely, for example from thin cards that can be connected like a puzzle. Four kinds of elements are enough.



From these elements, any path with 90-degree branches can be built. For example:





A route can be traversed automatically between the areas of each task (as in our project). Repeating three instructions in a good order is enough to crawl.

1) The robot follows the route (parameters: with which color sensor, which side of the route).

2) The robot turns to reach the intersection, which it detects with the other color sensor (parameter: in which direction).

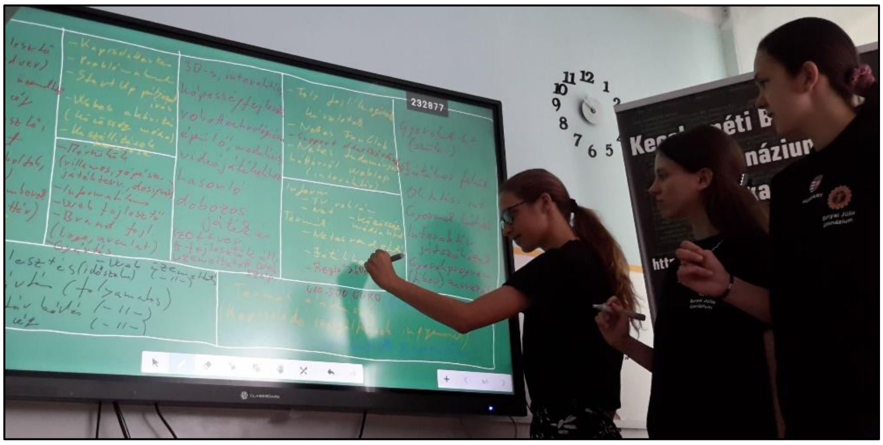
3) The robot goes straight ahead for some distance to pass through the intersection (parameter: distance).

This type of coding is simple enough for children to perform (without previous programming knowledge). As we have demonstrated in our project using Python code, the robot does not require a scripting system to function because it can be operated directly.

To turn our idea into a product that is genuinely enjoyable, serious electronics, game design, and mechanical development are needed. There is no need for LEGO-compliant hardware, as constructions can be built from much simpler technical elements. (Simple sensors, direct control, elegant constructions)

We like the idea. Maybe...

Additionally, we produced a Business Model Canvas for StartUp. Although we lack experience in this area, we have attempted.



Business Model Camvas

Key Partners

- Developer (hardware)
- Web operator
- Courier service
- Distribution network (e.g. toy stores)
- Professionals (scientific background)
- Research + Development

Key Activities

- Contact
- Problem management
- Writing applications for start-ups
- Web activities (social media)
- Delivery management

Key Resources

- Engineers (electrical engineer, mechanical engineer, product designers)
- Computer scientists
- Web developers
- Brand developers (logo, corporate identity)
- Production

Value Propositions

3D, interactive, skill-based, robotic technology modular game, similar to a video game and software, web service operated by developers

Customer Relationships

- Continuous improvement, additional stocks
- Fan Club on the Internet
- Support site (FAQ)
- Interactive website with scientific background

Channels

Information:

- Television commercials
- Internet
- Social media advertising

Product:

- Online order + courier service
- Toy stores

Customer Segments

- Children (6 years and older) - their parents
- Playful adults
- Educational institutions
- Organizers of children's programs (e.g. camps)
- Children's hospitals
- Interactive playhouses

Cost Structure

- Development (periodic)
- Warehouse rental
- Courier service
- Web operation

Revenue Streams

- Product sales (related services are free)

  A rate of about
  400-500 EUR/Base Kit
  200-300 EUR/Additional kits

- Royalties (for accessories produced by other companies)
- Brand sales

LIST OF SOURCES

[1] https://folyoiratok.oh.gov.hu/uj-pedagogiai-szemle/szamitogepes-jatekhasznalat-a-szocialis-kompetenciak-tukreben

[2] http://www.egeszsegmagazin.com/friss-hirek/20150504-videojatekokkal-a-szurkeallomanyert

[3] https://www.webmd.com/children/news/20181015/what-kids-fear-most-at-the-doctors-office

[4] https://www.researchgate.net/publication/38092244_Child-reported_hospital_fears_in_4_to_6-year-old_children

[5] https://www.youtube.com/watch?v=08Pb-UZPLiU

[6] https://www.youtube.com/watch?v=Xe817Aj-mgM 1:00-2:07

[7] https://www.verywellmind.com/left-brain-vs-right-brain-2795005

[8] https://www.healthline.com/health/left-brain-vs-right-brain

[9] https://matplotlib.org/

[10] https://ev3-dc.readthedocs.io/en/latest/