

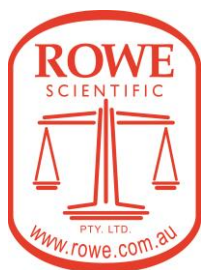


Prize Winner

**Programming, Apps &
Robotics
Year 3-4**

**Eric Wang
Michael Wang**

St Peter's College



DJI Battle

By Eric wang, Michael wang

Cover Sheet

COVER SHEET

PLEASE INCLUDE THIS COVER SHEET AS THE FIRST PAGE OF YOUR PROJECT FILE



Reports, risk assessments and logbooks (where applicable for the latter two) for Crystal Investigation and Models & Invention Reports OR Games (supporting videos for submission ONLY) due to be uploaded to our website (www.oliphantscienceawards.com.au) from 24 to 28 July 2023.

ID: 0680-032	St Peter's College
	Coordinator: Lisa Zollo School Phone: 08 8404 0400

Student	Gender	Patent: N
Eric Wang	M	
Michael Wang	M	

Year Level: 3-4 Group Entry: Y Students: 2
Category: Programming, Apps & Robotics Keycode: LZ5UN8QT
Title: DJI Battle

509-5093





Aim of entry and its scientific purpose

Our aim of the entry was to see how much the program can do, to see how much more the program can do than humans controlling the robot and how much difference there is between human control and program control.

How we got the idea

At the start we thought about using lego spike or EV3, but I wanted to use something new, so we then thought of things that we can do with the DJI S1. Our first ideas were doing a battle or shooting targets. We thought that doing a battle was more challenging so we chose to do that.



Experiencing the program.

At first it looked familiar but it turned out to be harder level than spike and EV3.



Constructing the S1

We built the S1 with our mom, when constructing it was very exciting and it made us feel like an engineer. At the time we just played a lot of battles without program. It was very fun.





System



LED Effects



Chassis



Gimbal



Blaster

First try

At the start we weren't very confident and we had to watch some video courses to help us. This was the first program we tried to do. But it didn't work, the problem was it could not shoot. And this was also too complicated.

```

function Shoot
  set PID Yaw parameters to Kp 110 Ki 0 Kd 10
  set PID Pitch parameters to Kp 95 Ki 0 Kd 5
  set MarkerList to identified vision marker info
  if MarkerList contains item 1 of ShootList ? then
    repeat until total items for ShootList == 0
      set Position to the first item 1 of ShootList index of MarkerList
      set ID to item Position of MarkerList
      set Error_x to item Position + 1 of MarkerList - 0.5
      set Error_y to 0.5 item Position + 2 of MarkerList
      set PID Yaw error to Error_x
      set PID Pitch error to Error_y
      set gimbal to rotate around yaw axis at PID Yaw output %/s and pitch axis at PID Pitch output %/s
  end if
end function

```

```

start
  set travel mode to free mode
  enable vision marker identification
  set gimbal rotation speed to 180 %/s
  set gimbal to rotate right in 90
  Save
  set gimbal to rotate left in 90
  Shoot
  disable vision marker identification
  turn on launch trajectory light
end start

```

```

function Save
  play sound scanning until finished
  set StartFlag to 0
  add 0 to the end of MarkerList
  repeat until StartFlag == 1
    repeat 20
      set MarkerList_2 to identified vision marker info
      if item 1 of MarkerList_2 > item 1 of MarkerList then
        set MarkerList to MarkerList_2
      end if
    end repeat
    wait 0.1 s
  end repeat
  if item 1 of MarkerList >= 1 then
    set i to 2
    repeat item 1 of MarkerList
      add item i of MarkerList to the end of ShootList
      increase i by 5
    end repeat
    set StartFlag to 1
  end if
end function

```




System



LED Effects



Chassis



Gimbal



Blaster



Extension

```

when S1 robot identified
  Eric2

```

```

start
  set chassis translation speed to 0.5 m/s
  set chassis rotation speed to 140 °/s
  repeat until S1 robot identified
    set chassis to translate forward at 15 degree(s) and rotate right
    wait 2.5 s
    set chassis to translate forward at -15 degree(s) and rotate left
    wait 2.5 s

```

```

function Eric2
  set travel mode to gimbal lead mode
  enable S1 robot identification
  wait until S1 robot identified
  repeat 15
    emit single infrared beam

```

Second try

This time we went through all the blocks so we can get more familiar with all of the blocks and know what they all can do. Also we decided to go more simple. This time we found some important things like enable s1 robot identification so the robot can actually detect the other robot. This time we also made a new function which I called Eric 2. we also made the robot move around so it won't be easy for the other robot to detect it. There wasn't a problem with this program we just needed to add more things.

Third try

This time we added a new function that is called eric4 this function is for what to do after been shot. But when we looked at the outcome the robot didn't react after been shot.

```
start
  enable S1 robot identification
  set chassis translation speed to 0.5 m/s
  set chassis rotation speed to 140 °/s
  repeat until S1 robot identified
    set chassis to translate forward at 15 degree(s) and rotate right
    wait 2.5 s
    set chassis to translate forward at -15 degree(s) and rotate left
    wait 2.5 s

when robot is hit by infrared beam
  Eric4

function Eric2
  set travel mode to gimbal lead mode
  enable S1 robot identification
  wait until S1 robot identified
  repeat 15
    emit single infrared beam

function Eric4
  set chassis rotation speed to 600 °/s
  set chassis translation speed to 3.5 m/s
  wait for hit on random armor
  set chassis to translate at 90 ° for 1 s
  set chassis to translate at 0 °
```

Fourth try

This time it worked, instead of making a function we just made “when random armor hit” then it will move away.

The screenshot displays the LEGO Mindstorms EV3 software interface, showing a Scratch-style block-based programming environment. The interface includes a left sidebar with categories: System, LED Effects, Chassis, Gimbal, Blaster, and Extension Module. The main workspace contains three scripts:

- When S1 robot identified:** A script block containing a function block named "Eric2".
- When random armor hit:** A script block containing four "set chassis to rotate/translate" blocks:
 - set chassis to rotate right for 0.5 s
 - set chassis to translate at 0 * for 0.5 s
 - set chassis to rotate left for 0.5 s
 - set chassis to translate at 0 * for 3 s
- Start:** A script block containing:
 - enable S1 robot identification
 - set chassis translation speed to 0.5 m/s
 - set chassis rotation speed to 140 °/s
 - repeat until S1 robot identified:
 - set chassis to translate forward at 15 degree(s) and rotate right
 - wait 2.5 s
 - set chassis to translate forward at -15 degree(s) and rotate left
 - wait 2.5 s

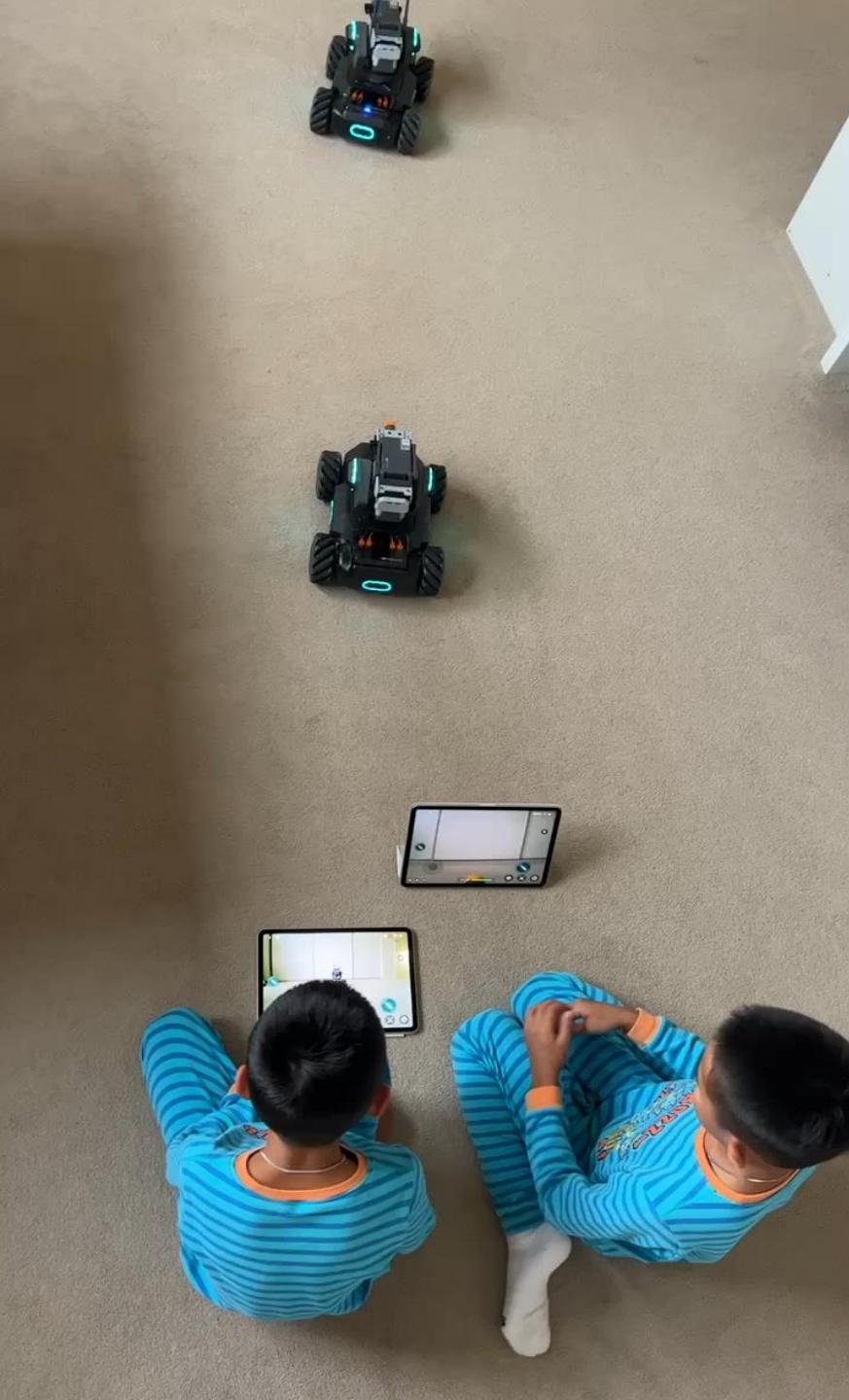
A separate function block named "Eric2" is defined with the following blocks:

- set travel mode to gimbal lead mode
- enable S1 robot identification
- wait until S1 robot identified
- repeat 20:
 - emit single infrared beam



Video part 1

This what happens when the robot detects the other robot it will shoot.



Video part 2

This what happens when the robot gets shot. The robot tries to dodge the beams.

Problems

When we make all of the program happen, there was 2 problems

1. After it shoots it will not go back to moving again, instead it just stops the programs.
2. When it gets shot the robot does not react.



New Program/fifth try

My solution to both problems worked, so the solution for 1st problem was solved by adding a function which I called "Messi" I will add that to what happens after being shot or after s1 identified. The problem for the 2nd problem was solved when I noticed I put "when random armor hit" I remembered that I used infrared beams so I changed the block to "when robot hit by infrared beam". (Eric)

The screenshot displays a block-based programming environment with a sidebar on the left containing the following categories: System, LED Effects, Chassis, Gimbal, Blaster, and Extension Module. The main workspace features a 'function Messi' block and two event-driven blocks. The 'function Messi' block contains the following sequence of actions: 'enable S1 robot identification', 'set chassis translation speed to 0.5 m/s', 'set chassis rotation speed to 50 °/s', a 'repeat until' loop with conditions 'robot hit by infrared beam' or 'S1 robot identified', 'set chassis to translate forward at 15 degree(s) and rotate right', 'wait 2.5 s', 'set chassis to translate forward at -15 degree(s) and rotate left', and 'wait 2.5 s'. The 'when S1 robot identified' event block includes: 'set chassis to stop moving', a 'repeat 10' loop containing 'emit single infrared beam' and 'wait 0.1 s', and 'Messi'. The 'when robot is hit by infrared beam' event block includes: 'set chassis to stop moving', 'set chassis to translate at -90 ° for 1 s', 'set chassis to translate at 90 ° for 1 s', and 'Messi'. A 'start' block at the top right contains 'enable S1 robot identification' and 'Messi'. A 'Connect' button is located in the top right corner of the interface.

Another Problem

Another problem was after it attempts to shoot once and goes back to doing messi it can't detect the other robot again, so we changed by adding an always block around "Messi" in the code at "start"

The screenshot displays a block-based programming environment with a sidebar on the left containing icons for System, LED Effects, Chassis, Gimbal, Blaster, and Extension Module. The main workspace shows the following code blocks:

- start** block:
 - enable S1 robot identification
 - always loop containing a **Messi** block
 - repeat until loop with conditions "robot hit by infrared beam" or "S1 robot identified":
 - set chassis translation speed to 0.5 m/s
 - set chassis rotation speed to 140 °/s
 - set chassis to translate forward at 15 degree(s) and rotate right
 - wait 2.5 s
 - set chassis to translate forward at -15 degree(s) and rotate left
 - wait 2.5 s
- when robot is hit by infrared beam** block:
 - set chassis to stop moving
 - set chassis translation speed to 1 m/s
 - set chassis to translate at 0 ° for 1 s
 - set chassis to translate at 180 ° for 1 s
 - Messi
- when S1 robot identified** block:
 - repeat 5 loop:
 - emit single infrared beam
 - wait 0.1 s
 - Messi

But...

But if my robot defeated the other robot mines will keep on shooting if it is in its sight. So I removed a block from when s1 robot identified, I removed the block set chassis to stop moving, so when my robot identifies the enemy robot it will keep moving so my robot will stop continuously defeating the enemy.(Eric)

The screenshot displays the LEGO Mindstorms software interface with a dark background. On the left is a vertical toolbar with icons for System, LED Effects, Chassis, Gimbal, Blaster, and Extension Module. The main workspace contains several block-based programs:

- Start Program:** A blue 'start' block followed by an orange 'enable S1 robot identification' block, an 'always' loop block, and a 'Messi' function block.
- Messi Function:** A blue 'function Messi' block containing:
 - An orange 'enable S1 robot identification' block.
 - A blue 'set chassis translation speed to 0.5 m/s' block.
 - A blue 'set chassis rotation speed to 140 °/s' block.
 - A yellow 'repeat until' block with conditions 'robot hit by infrared beam' and 'or S1 robot identified'.
 - A blue 'set chassis to translate forward at 15 degree(s) and rotate right' block.
 - A blue 'wait 2.5 s' block.
 - A blue 'set chassis to translate forward at -15 degree(s) and rotate left' block.
 - A blue 'wait 2.5 s' block.
- When robot is hit by infrared beam:** A yellow 'when robot is hit by infrared beam' block containing:
 - A blue 'set chassis to stop moving' block.
 - A blue 'set chassis translation speed to 1 m/s' block.
 - A blue 'set chassis to translate at 0 ° for 1 s' block.
 - A blue 'set chassis to translate at 180 ° for 1 s' block.
 - A blue 'Messi' block.
- When S1 robot identified:** An orange 'when S1 robot identified' block containing:
 - A blue 'repeat 5' block.
 - An orange 'emit single infrared beam' block.
 - A blue 'wait 0.1 s' block.
 - A blue 'Messi' block.

At the top right, there are navigation icons (back, forward, refresh) and a blue 'Connect' button.



Navigation icons: back, undo, redo, home, left arrow, right arrow, and a blue "Connect" button.

- System
- LED Effects
- Chassis
- Gimbal
- Blaster
- Extension Module

```
function Messi
  enable S1 robot identification
  set chassis translation speed to 0.5 m/s
  set chassis rotation speed to 140 */s
  repeat until robot hit by infrared beam or S1 robot identified
    set chassis to translate forward at 15 degree(s) and rotate right
    wait 2.5 s
    set chassis to translate forward at -15 degree(s) and rotate left
    wait 2.5 s
  repeat

start
  enable S1 robot identification
  always
  Messi

when robot is hit by infrared beam
  set chassis to stop moving
  set chassis translation speed to 1 m/s
  set chassis to translate at 0 * for 1 s
  set chassis to translate at 180 * for 1 s
  Messi

when S1 robot identified
  repeat 5
    emit single infrared beam
    wait 0.1 s
  Messi
```

Success

The final product was a success it worked.



System



LED Effects



Chassis



Gimbal



Blaster



```

function Messi
  enable S1 robot identification
  set chassis translation speed to 0.5 m/s
  set chassis rotation speed to 140 °/s
  repeat until robot hit by infrared beam or S1 robot identified
    set chassis to translate forward at 15 degree(s) and rotate right
    wait 2.5 s
    set chassis to translate forward at -15 degree(s) and rotate left
    wait 2.5 s
  
```

```

start
  enable S1 robot identification
  always
    Messi
  
```

```

when robot is hit by infrared beam
  set chassis to stop moving
  set chassis translation speed to 1 m/s
  set chassis to translate at 0 ° for 1 s
  set chassis to translate at 180 ° for 1 s
  Messi
  
```

```

when S1 robot identified
  repeat 5
    emit single infrared beam
    wait 0.1 s
  Messi
  
```

Program

There are 4 parts to our program.

Part 1

This part of the program is a function called "messi". This function controls how the robot moves. I have chosen to make the robot move in a figure 8. The first bit is "enable s1 robot identification" so the program allows the robot to identify the other s1. Then the next 2 blocks is controlling the speed translation speed is 0.5 m/s while the rotation speed is 70 degrees/s, next is when the robot actually moves. "set chassis to translate forward at 15 degrees and rotate right" and "wait 5 s" will make the robot run half of the figure eight. The other half will run when I add "set chassis to translate forward at -15 degrees and rotate left" and "wait 5 s". I've put this part of the program in a "repeat until robot hit by infrared beam or s1 robot identified" so that this will repeat until hit by infrared beam or s1 robot identified.

The screenshot displays a block-based programming environment with a dark background. On the left is a vertical toolbar with icons for System, LED Effects, Chassis, Gimbal, Blaster, and Extension Module. The main workspace shows a function block named "Messi" with the following sequence of blocks:

- An orange "enable" block with "S1 robot" selected in the dropdown and "identification" in the text field.
- A purple "set chassis translation speed to" block with "0.5" in the input field and "m/s" in the unit dropdown.
- A purple "set chassis rotation speed to" block with "70" in the input field and "°/s" in the unit dropdown.
- A blue "repeat until" loop block with two conditions: "robot hit by infrared beam" and "S1 robot identified", connected by an "or" operator.
- Inside the loop, a purple "set chassis to translate forward at" block with "15" in the input field, "degree(s)" in the unit dropdown, and "right" in the rotation dropdown.
- A grey "wait" block with "5" in the input field and "s" in the unit dropdown.
- A purple "set chassis to translate forward at" block with "-15" in the input field, "degree(s)" in the unit dropdown, and "left" in the rotation dropdown.
- A grey "wait" block with "5" in the input field and "s" in the unit dropdown.
- A blue "repeat" icon at the end of the loop block.

At the top right of the interface, there are navigation icons (back, forward, home, play) and a blue "Connect" button.

Part 2

In this part this what happens when the program starts, first it is "enable s1 robot identification" so the program allows the robot to identify the other robot. Then the program does the function "messi" (slide 17) the "messi" will be put in an always so the program won't stop.

The screenshot shows a programming environment with a dark background. On the left is a vertical sidebar with icons and labels for different robot components: System (blue robot icon), LED Effects (green bell icon), Chassis (purple robot icon), Gimbal (red arrow icon), Blaster (orange target icon), and Extension Module (pink person icon). The main workspace contains a blue 'start' block. Attached to it is an orange block with three fields: 'enable', 'S1 robot', and 'identification'. Below this is a blue 'always' loop block, which contains a blue 'Messi' block. At the top right of the workspace are navigation icons: a back arrow, two circular arrows (undo/redo), a monitor icon, a code icon, and a play icon. A blue 'Connect' button is also visible in the top right corner.

Part 3

In this part it controls what happens when the other robot is identified. So the first block is "when s1 robot identified" then "emit single infrared beam" and then "wait 0.1s" this is for shooting, this part will be put into a "repeat 5" then after all of that the program goes back to the function messi.

The screenshot displays a block-based programming environment. On the left, a vertical sidebar lists several categories: System, LED Effects, Chassis, Gimbal, Blaster, and Extension Module. The main workspace features a script starting with a 'when S1 robot identified' trigger block. This is followed by a 'repeat 5' loop block. Inside the loop, there are three blocks: 'emit single infrared beam', 'wait 0.1 s', and a 'Messi' block. The 'Messi' block is a blue block with a white outline. The interface also includes navigation icons at the top (back, forward, and a 'Connect' button) and a dark background.

Part 4

This part of the program is to control what to do when been hit. The first block is “when robot is hit by infrared beam” then “set chassis to stop moving” so the robot could stop at were it is and prepare for the next part of the program. After stopping the next block will be “set chassis translation speed to 1 m/s” that will change the speed. Next block will be “ set chassis to translate at 0 degrees for 1s” this block is used for moving forward. So next we have “set chassis to translate at 180 degrees for 1s” so the robot can move back to its original position. After that it will go to “messi”.

System

LED Effects

Chassis

Gimbal

Blaster

Extension Module

Connect

when robot is hit by infrared beam

- set chassis to stop moving
- set chassis translation speed to 1 m/s
- set chassis to translate at 0 ° for 1 s
- set chassis to translate at 180 ° for 1 s

Messi

The second robot

This is the robot competing against the robot that we just programmed.

The image shows a Scratch code editor interface with a dark background. At the top, there are navigation icons: a back arrow, a refresh icon, a code editor icon, a play button, and a 'Connect' button with a robot icon. On the left side, there is a vertical toolbar with icons for 'stem', 'Effects', 'chassis', 'mbal', 'aster', and 'ension module'. The main workspace contains four code blocks:

- start** block: Contains 'enable S1 robot identification', 'set travel mode to chassis lead mode', and an 'always' loop containing a 'CR7' block.
- function CR7** block: Contains 'enable S1 robot identification', 'set chassis translation speed to 0.5 m/s', a 'repeat until' loop with conditions 'S1 robot identified' and 'robot hit by infrared beam', 'set chassis to translate forward at 5 degree(s) and rotate right', and 'wait 15 s'.
- when S1 robot identified** block: Contains a 'repeat 5' loop with 'emit single infrared beam' and 'wait 0.1 s'.
- when robot is hit by infrared beam** block: Contains 'set chassis to stop moving', 'set chassis translation speed to 1 m/s', 'set chassis to translate at 0 ° for 1 s', 'set chassis to translate at 180 ° for 1 s', and a 'CR7' block.

Second robot program.

In this program the only difference from the first robot is the movement.

Movement of second robot program.

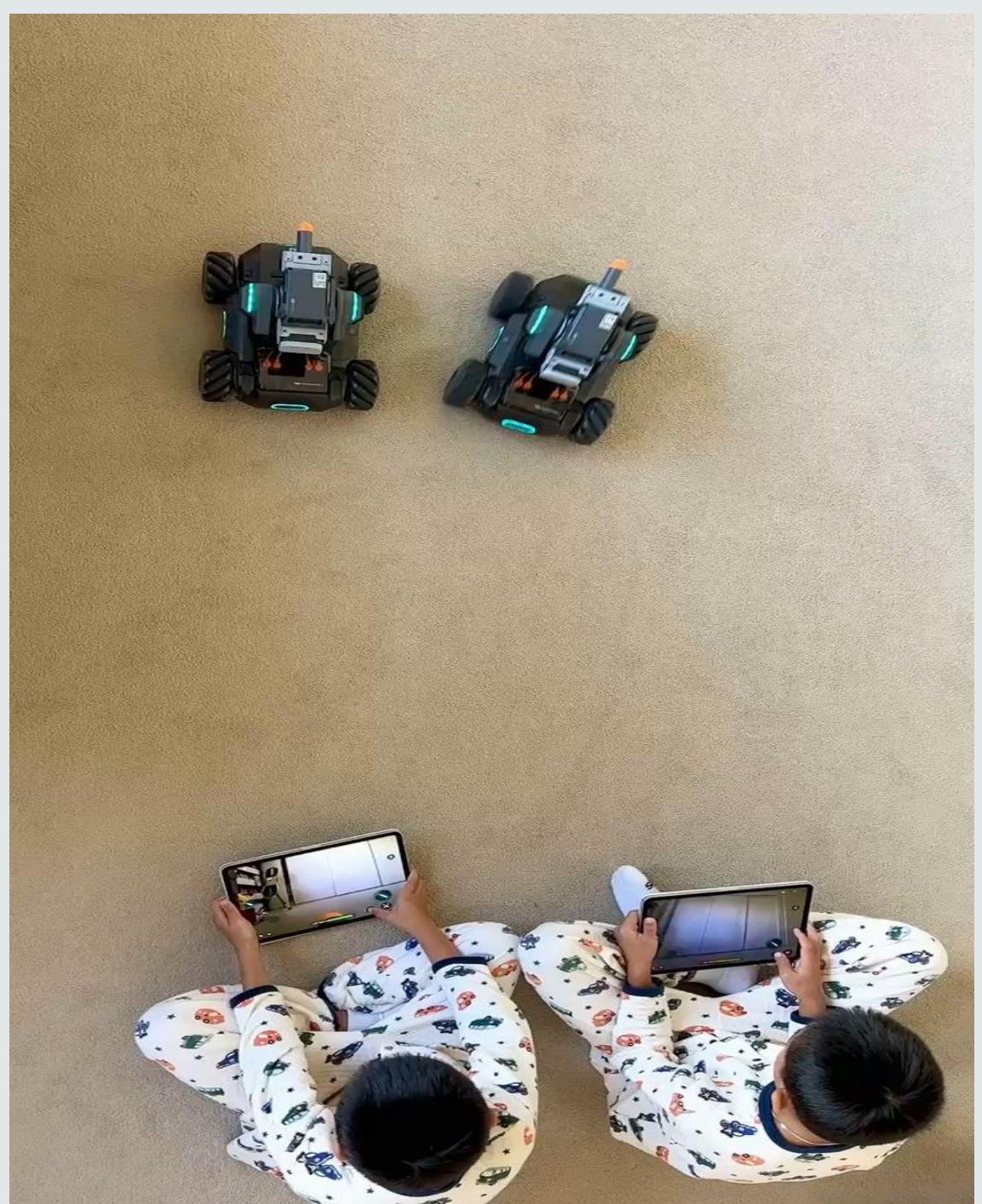
First there will be a function called CR7 instead of the first robot's function name messi. Next block will be 'enable s1 robot identification' so the program allows the robot to identify the other robot. After that we add a block "set translation speed to 0.5 m/s" so we can make the speed perfect. Next we add the block "set chassis to translate forward at 5 degrees and rotate right" and " wait 15 s" that part was for moving in a circle movement. That part of the program will be wrapped in a "repeat until s1 robot identified or robot hit by infrared beam".

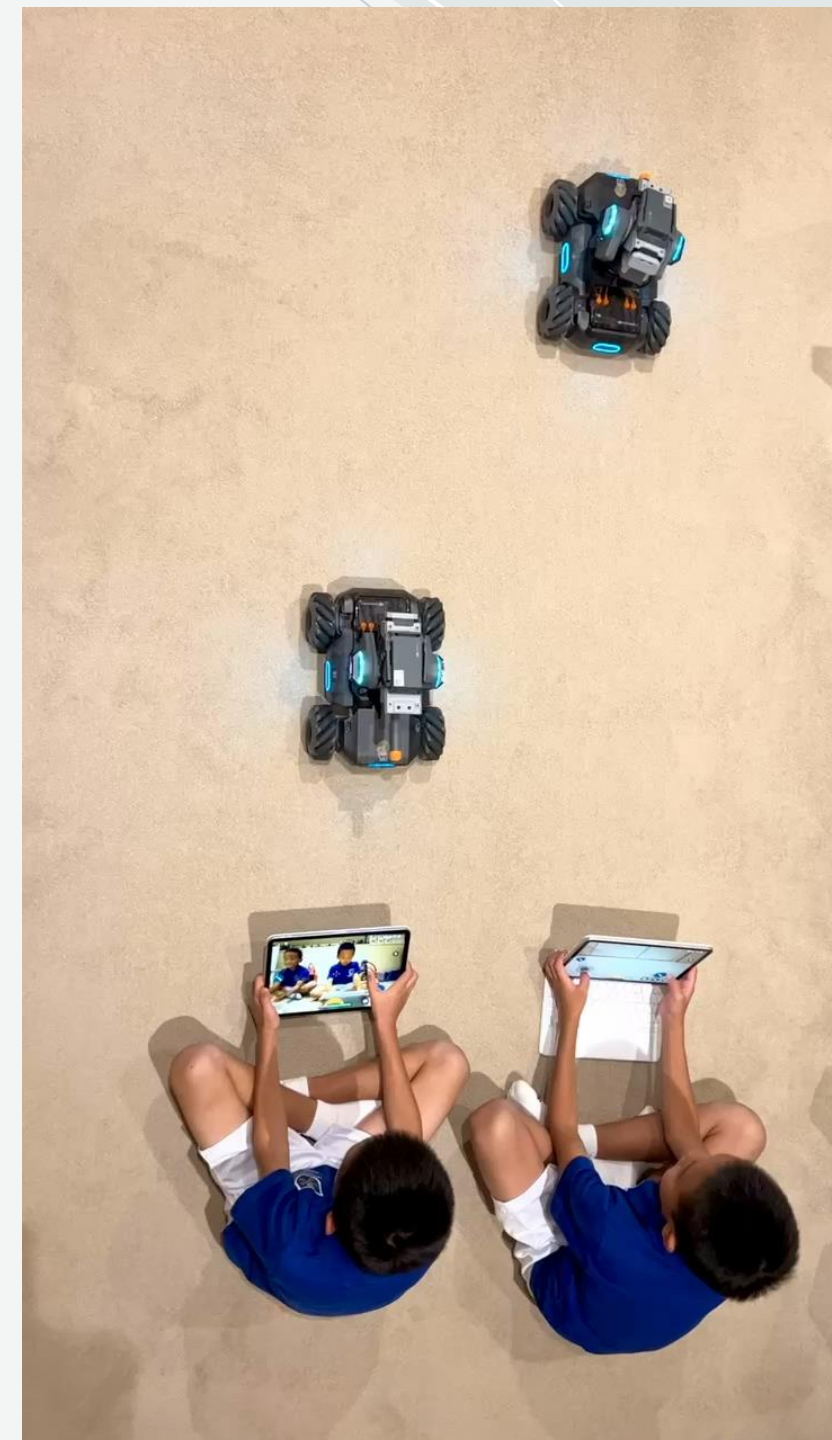
The image shows a block-based programming environment. On the left is a sidebar with icons and labels for different system components: System, LED Effects, Chassis, Gimbal, Blaster, and Extension Module. At the top right, there are navigation icons (back, forward, home) and a 'Connect' button. The main workspace displays a function block named 'CR7'. Inside this function, the following blocks are stacked:

- 'enable S1 robot identification' (orange block)
- 'set chassis translation speed to 0.5 m/s' (purple block)
- 'repeat until' loop (blue block) with two conditions: 'S1 robot identified' (orange) and 'robot hit by infrared beam' (yellow), connected by 'or'.
- 'set chassis to translate forward at 5 degree(s) and rotate right' (purple block)
- 'wait 15 s' (blue block)

The entire function is enclosed in a blue frame with a refresh icon at the bottom right.

The finishing battle video



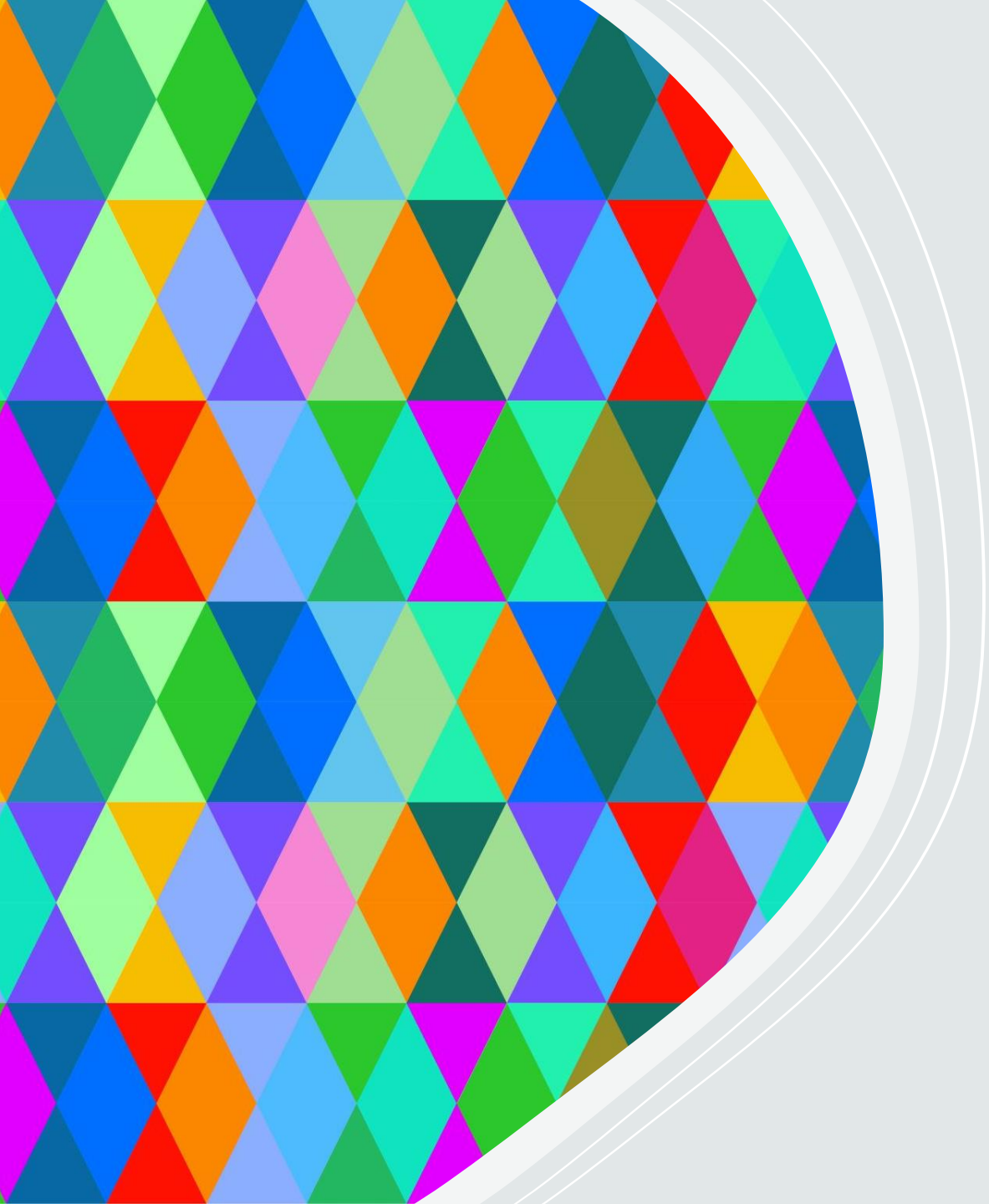


Human control

This is two robots fighting using human control.

Conclusion

1. How much the program can do: how much the program can do depends on how much creativity the human has.
2. How much more the program can do more than humans: we think sometimes one program can do more than one human, like our DJI robot if we have many then one program can make them all work at the same time while if it is human controlled then many robots will need many humans, and a program you can make it repeat almost forever by changing batteries, but humans can feel tired and needs to rest for a long time.
3. How much difference there is between program control and human control: the main difference is that the human would be influenced by their emotions, emotions and feelings can influence the human's decision, but the program does not have this problem, like the two videos before, we experimented in the same sized area, and when my robot hit my partner's robot he felt nervous and was not sure of what to do with his robot so his robot got defeated first, but in the program controlled video my partner's robot won! Interesting, in human controlled battles i usually win but when we use program controlled my partner wins.



Thank you