

AI for STEM Competition

Asia Pacific STEAM_AI Technology
Innovation Challenge

06 Barrel Launch (Part II)

CocoRobo





Chapter 1
Improve
optimization
analytics



Chapter 3
Projection
optimization



Chapter 2
Speed optimization

目 録
CONTENTS

ONE.

Improve
optimization
analytics

Improve optimization analytics

Think about it, what can be optimized in the handling of the car?

For example: velocity switching, firing intensity, etc





Improve optimization analytics



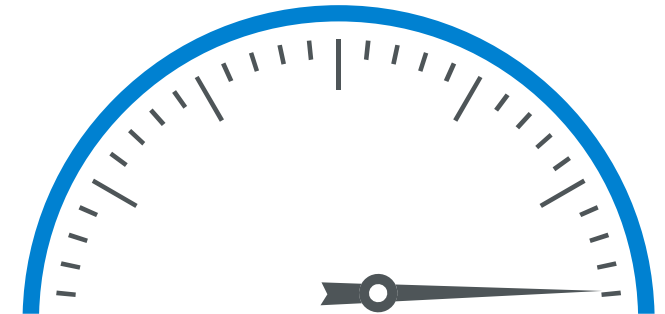
low speed

Speed 60



Medium speed

Speed 150



high speed

Speed 240

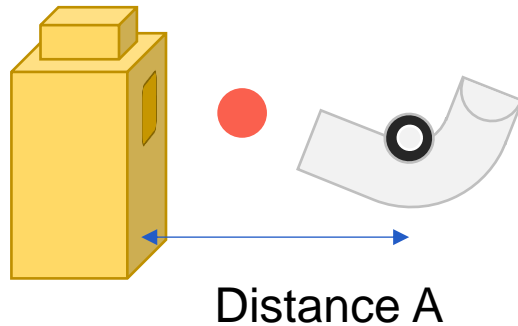
The faster the speed, the faster it moves, and the more powerful it is

The angle of servo motor P0 is 60 degrees, and the speed of motor E is 60

Improve optimization analytics

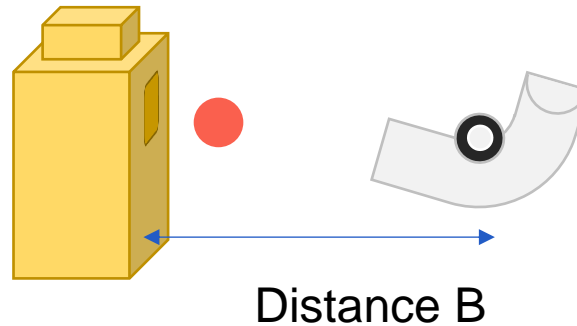
Self-scaling method:

Think about it, is there any other way to optimize?



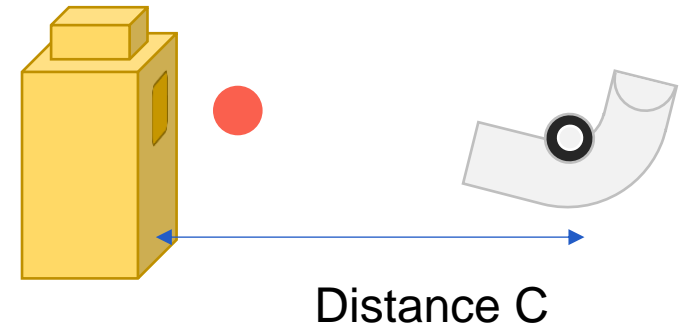
Mode 1

The angle of servo motor P0 is 60 degrees,
And the speed of motor E is 60



Mode 2

The angle of servo motor P0 is 40 degrees,
and the speed of motor E is 150



Mode 3

The angle of servo motor P0 is 20 degrees,
and the speed of motor M2 is 240

TWO.

Speed optimization

● Speed optimization



Task 1: Test the 2 or 3 speed values you need and record them

	low speed	Medium speed	high speed
Speed value			

● Speed optimization

Think about it, how to realize the switching of different speeds of the trolley?



● Speed optimization

A Scratch 'Set speed to 200' block. It consists of an orange 'Set' block, a dropdown menu showing 'speed', an equals sign, and a light blue numeric input field containing '200'.A Scratch 'Set speed to 100' block. It consists of an orange 'Set' block, a dropdown menu showing 'speed', an equals sign, and a light blue numeric input field containing '100'.

The new variable speed is added, and different speed values are assigned in different cases

Speed optimization

 **Task 2:** Set **SQUARE** button as medium speed, **CIRCLE** as low speed

Results



Send a
message
"SQUARE"

Set speed = 200



Send a
message "
CIRCLE"

Set speed = 100

Set the speed value
of each function of
the trolley movement
to the variable speed

Step 1

Step 2

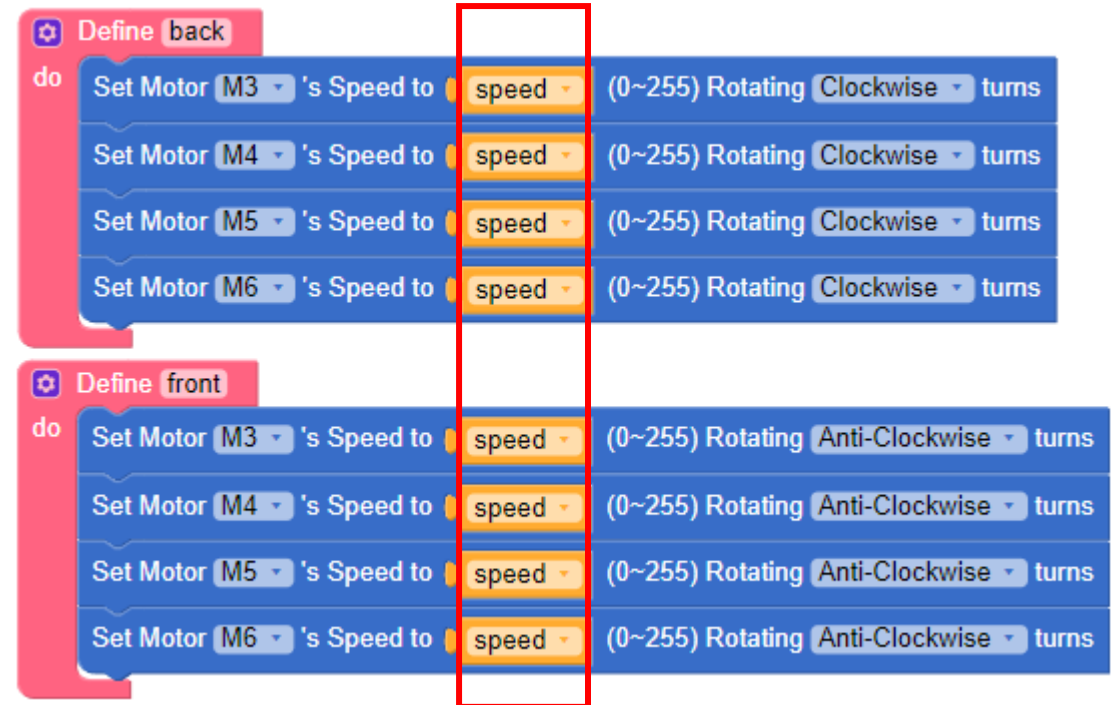
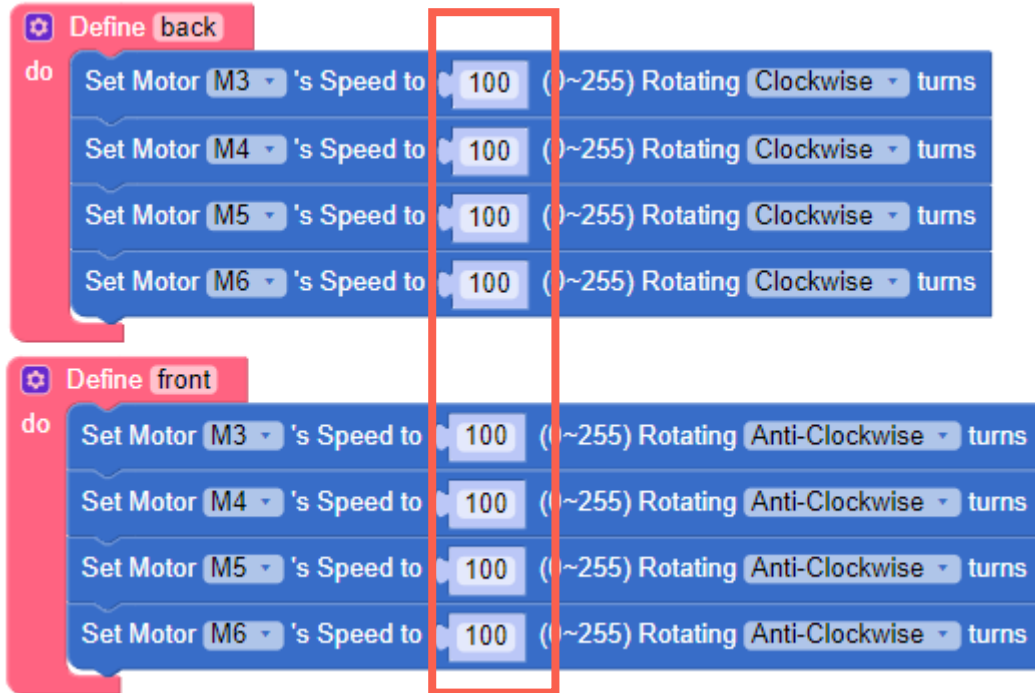
Step 3

Speed optimization



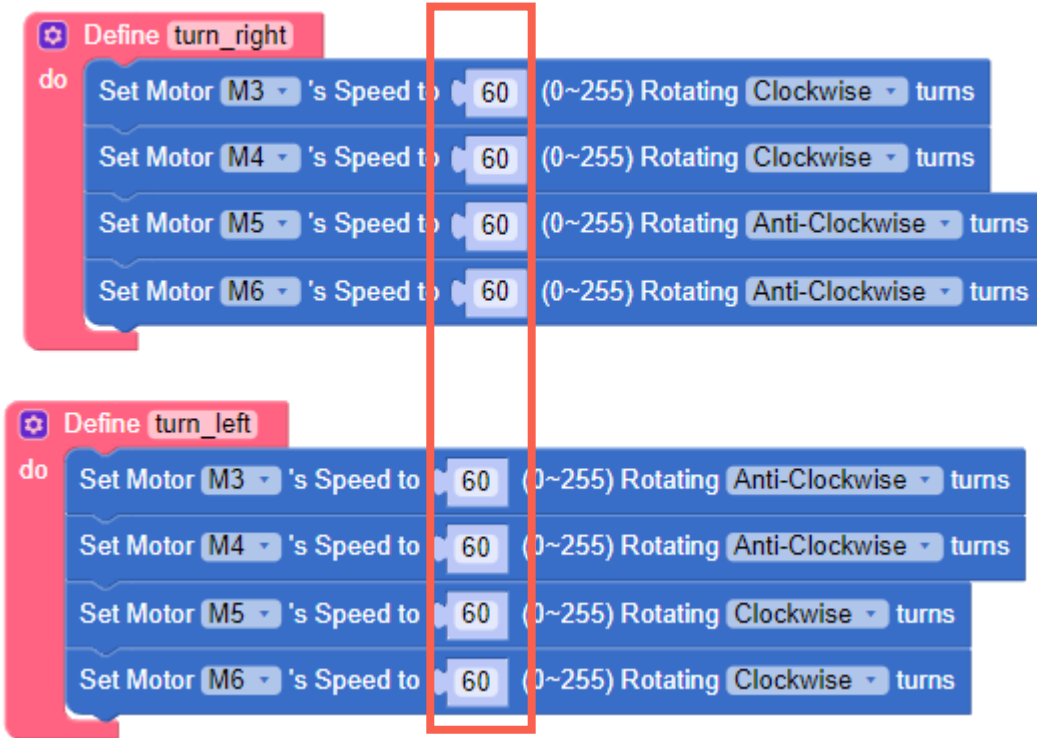
Task 2: Set **SQUARE** button as medium speed, **CIRCLE** as low speed

Use the speed variable instead of the value



Speed optimization

 **Task 2:** Set **SQUARE** button as medium speed, **CIRCLE** as low speed



Turnleft and Turnright are used for micro-adjustment when projecting, and only need to be executed at low speed, and there is no need to set different speed values.

Speed optimization

Reference



Task 2: Set **SQUARE** button as medium speed, **CIRCLE** as low speed

```

Motor Driver Setup
Set Motor M2's Speed to 0 (0~255) Rotating Clockwise turns
Set Motor M3's Speed to 0 (0~255) Rotating Clockwise turns
Set Motor M4's Speed to 0 (0~255) Rotating Clockwise turns
Set Motor M5's Speed to 0 (0~255) Rotating Clockwise turns
Set Motor M6's Speed to 0 (0~255) Rotating Clockwise turns

Servo Setup
Set Servo on GPIO # P0 Rotate to 0 Degree (0°~180°)
Set Servo on GPIO # P1 Rotate to 90 Degree (0°~180°)
Set Servo on GPIO # P2 Rotate to 90 Degree (0°~180°)
Set Servo on GPIO # P3 Rotate to 0 Degree (0°~180°)

UART
Obtain data through the serial port to initialize
Set Baud Rate: 115200 bps
Set Data = ""
Set control = 100
Set state = 0
Set speed = 100

Repeat forever
Do Clear serial port cache data
try
Set Data = Get Serial (UART) Data at 0 as
except
Set control = Split string to a list (Data) by delimiter: Split and generate a list
If List (control) # 0 item > 100
Do right
Else if List (control) # 1 item < -100
Do left
Else if List (control) # 2 item > 100
Do back
Else if List (control) # 2 item < -100
Do front

```

```

Else if List (control) # 0 item < -100
Do turn_right
Else if List (control) # 1 item > 100
Do turn_left
Else if List (control) # 2 item > 100
Do shoot_down
Else if List (control) # 2 item < -100
Do shoot_up
Else
If Data == "" NONE
Do stop
Set state = 0
Else if Data == "" UP
Do Set Servo on GPIO # P1 Rotate to 180 Degree (0°~180°)
Else if Data == "" DOWN
Do Set Servo on GPIO # P1 Rotate to 14 Degree (0°~180°)
Else if Data == "" LEFT
Do Set Servo on GPIO # P1 Rotate to 0 Degree (0°~180°)
Else if Data == "" LT
Do Set Servo on GPIO # P2 Rotate to 120 Degree (0°~180°)
Else if Data == "" RT
Do Set Servo on GPIO # P2 Rotate to 90 Degree (0°~180°)
Else if Data == "" TRIANGLE
Do If state == 0
Do Set state = 1
shoot_ball
Else if Data == "" SQUARE
Do Set speed = 200
Else if Data == "" CIRCLE
Do Set speed = 100

```

```

Define right
do
Set Motor M3's Speed to speed (0~255) Rotating Clockwise turns
Set Motor M4's Speed to speed (0~255) Rotating Anti-Clockwise turns
Set Motor M5's Speed to speed (0~255) Rotating Clockwise turns
Set Motor M6's Speed to speed (0~255) Rotating Anti-Clockwise turns

```

```

Define left
do
Set Motor M3's Speed to speed (0~255) Rotating Anti-Clockwise turns
Set Motor M4's Speed to speed (0~255) Rotating Clockwise turns
Set Motor M5's Speed to speed (0~255) Rotating Anti-Clockwise turns
Set Motor M6's Speed to speed (0~255) Rotating Clockwise turns

```

```

Define back
do
Set Motor M3's Speed to speed (0~255) Rotating Clockwise turns
Set Motor M4's Speed to speed (0~255) Rotating Clockwise turns
Set Motor M5's Speed to speed (0~255) Rotating Clockwise turns
Set Motor M6's Speed to speed (0~255) Rotating Clockwise turns

```

```

Define front
do
Set Motor M3's Speed to speed (0~255) Rotating Anti-Clockwise turns
Set Motor M4's Speed to speed (0~255) Rotating Anti-Clockwise turns
Set Motor M5's Speed to speed (0~255) Rotating Anti-Clockwise turns
Set Motor M6's Speed to speed (0~255) Rotating Anti-Clockwise turns

```

```

Define shoot_up
do
If shoot < 70
Do Change shoot by 5
Else
Set shoot = 70
Set Servo on GPIO # P0 Rotate to shoot Degree (0°~180°)

```

```

Define turn_right
do
Set Motor M3's Speed to 60 (0~255) Rotating Clockwise turns
Set Motor M4's Speed to 60 (0~255) Rotating Clockwise turns
Set Motor M5's Speed to 60 (0~255) Rotating Anti-Clockwise turns
Set Motor M6's Speed to 60 (0~255) Rotating Anti-Clockwise turns

```

```

Define turn_left
do
Set Motor M3's Speed to 60 (0~255) Rotating Anti-Clockwise turns
Set Motor M4's Speed to 60 (0~255) Rotating Anti-Clockwise turns
Set Motor M5's Speed to 60 (0~255) Rotating Clockwise turns
Set Motor M6's Speed to 60 (0~255) Rotating Clockwise turns

```

```

Define shoot_ball
do
Set Motor M2's Speed to 200 (0~255) Rotating Clockwise turns
Set Servo on GPIO # P3 Rotate to 180 Degree (0°~180°)
Wait 1000 Milliseconds
Set Servo on GPIO # P3 Rotate to 0 Degree (0°~180°)
Wait 500 Milliseconds
Set Motor M2's Speed to 0 (0~255) Rotating Clockwise turns

```

```

Define stop
do
Set Motor M3's Speed to 0 (0~255) Rotating Clockwise turns
Set Motor M4's Speed to 0 (0~255) Rotating Clockwise turns
Set Motor M5's Speed to 0 (0~255) Rotating Clockwise turns
Set Motor M6's Speed to 0 (0~255) Rotating Clockwise turns

```

```

Define shoot_down
do
If shoot > 0
Do Change shoot by -5
Else
Set shoot = 0
Set Servo on GPIO # P0 Rotate to shoot Degree (0°~180°)

```

P

O

W

E

—

THREE.


Projection optimization

●

C H A P T E R

●

Projection optimization

 **Task 3:** Setup **TRIANGLE** button for high-speed launch, **CROSS** for low speed launch

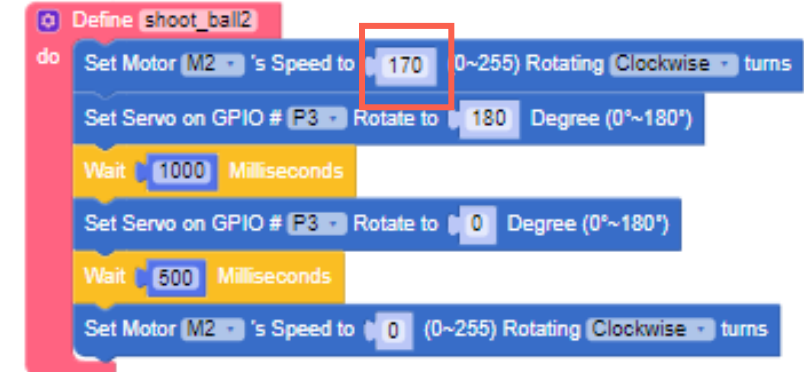
Result



Send a
message "TRIANGLE"



Send a
message "CROSS"



Step 1

Step 2

Projection optimization

Reference



Task 3: Setup **TRIANGLE** button for high-speed launch, **CROSS** for low speed launch

```

Motor Driver Setup
Set Motor [M2] 's Speed to 0 (0~255) Rotating Clockwise turns
Set Motor [M3] 's Speed to 0 (0~255) Rotating Clockwise turns
Set Motor [M4] 's Speed to 0 (0~255) Rotating Clockwise turns
Set Motor [M5] 's Speed to 0 (0~255) Rotating Clockwise turns
Set Motor [M6] 's Speed to 0 (0~255) Rotating Clockwise turns

Servo Setup
Set Servo on GPIO # [P0] Rotate to 0 Degree (0~180°)
Set Servo on GPIO # [P1] Rotate to 90 Degree (0~180°)
Set Servo on GPIO # [P2] Rotate to 90 Degree (0~180°)
Set Servo on GPIO # [P3] Rotate to 0 Degree (0~180°)

UART
Obtain data through the serial port to initialize
Set Baud Rate: 115200 bps
Set [Data] = ""
Set [control] = 0
Set [state] = 0
Set [speed] = 0
Repeat forever
Do
  Clear serial port cache data
  try
    Set [Data] = Get Serial (UART) Data at 0 as
  except
    Set [control] = Split string to a list [Data] by delimiter: ; Split and generate a list
  Do
    If [List [control] # 0 item <= 1]
    Do
      If [int [List [control] # 0 item] >= 100]
      Do
        right
      Else if [int [List [control] # 0 item] <= -100]
      Do
        left
      Else if [int [List [control] # 0 item] >= 100]
      Do
        back
      Else if [int [List [control] # 0 item] <= -100]
      Do
        front
  
```

```

Else if [int [List [control] # 0 item] >= 100]
Do
  turn_right
Else if [int [List [control] # 0 item] <= -100]
Do
  turn_left
Else if [int [List [control] # 0 item] >= 100]
Do
  shoot_down
Else if [int [List [control] # 0 item] <= -100]
Do
  shoot_up
Else
  If [Data == "NONE"]
  Do
    stop
    Set [state] = 0
  Else if [Data == "UP"]
  Do
    Set Servo on GPIO # [P1] Rotate to 180 Degree (0~180°)
  Else if [Data == "DOWN"]
  Do
    Set Servo on GPIO # [P1] Rotate to 14 Degree (0~180°)
  Else if [Data == "LEFT"]
  Do
    Set Servo on GPIO # [P1] Rotate to 90 Degree (0~180°)
  Else if [Data == "RIGHT"]
  Do
    Set Servo on GPIO # [P1] Rotate to 90 Degree (0~180°)
  Else if [Data == "L1"]
  Do
    Set Servo on GPIO # [P2] Rotate to 120 Degree (0~180°)
  Else if [Data == "R1"]
  Do
    Set Servo on GPIO # [P2] Rotate to 90 Degree (0~180°)
  Else if [Data == "TRIANGLE"]
  Do
    If [state == 0]
    Do
      Set [state] = 1
      shoot_ball
    Else if [Data == "CROSS"]
    Do
      If [state == 0]
      Do
        Set [state] = 1
        shoot_ball2
  Else if [Data == "SQUARE"]
  Do
    Set [speed] = 200
  Else if [Data == "CIRCLE"]
  Do
    Set [speed] = 100
  
```

```

Define [right]
do
  Set Motor [M3] 's Speed to [speed] (0~255) Rotating Clockwise turns
  Set Motor [M4] 's Speed to [speed] (0~255) Rotating Anti-Clockwise turns
  Set Motor [M5] 's Speed to [speed] (0~255) Rotating Clockwise turns
  Set Motor [M6] 's Speed to [speed] (0~255) Rotating Anti-Clockwise turns

Define [left]
do
  Set Motor [M3] 's Speed to [speed] (0~255) Rotating Anti-Clockwise turns
  Set Motor [M4] 's Speed to [speed] (0~255) Rotating Clockwise turns
  Set Motor [M5] 's Speed to [speed] (0~255) Rotating Anti-Clockwise turns
  Set Motor [M6] 's Speed to [speed] (0~255) Rotating Clockwise turns

Define [back]
do
  Set Motor [M3] 's Speed to [speed] (0~255) Rotating Clockwise turns
  Set Motor [M4] 's Speed to [speed] (0~255) Rotating Clockwise turns
  Set Motor [M5] 's Speed to [speed] (0~255) Rotating Clockwise turns
  Set Motor [M6] 's Speed to [speed] (0~255) Rotating Clockwise turns

Define [front]
do
  Set Motor [M3] 's Speed to [speed] (0~255) Rotating Anti-Clockwise turns
  Set Motor [M4] 's Speed to [speed] (0~255) Rotating Anti-Clockwise turns
  Set Motor [M5] 's Speed to [speed] (0~255) Rotating Anti-Clockwise turns
  Set Motor [M6] 's Speed to [speed] (0~255) Rotating Anti-Clockwise turns

Define [shoot_up]
do
  If [shoot <= 70]
  Do
    Change [shoot] by 5
  Else
    Set [shoot] = 70
    Set Servo on GPIO # [P0] Rotate to [shoot] Degree (0~180°)

Define [shoot_down]
do
  If [shoot >= 0]
  Do
    Change [shoot] by -5
  Else
    Set [shoot] = 0
    Set Servo on GPIO # [P0] Rotate to [shoot] Degree (0~180°)
  
```

```

Define [turn_right]
do
  Set Motor [M3] 's Speed to 60 (0~255) Rotating Clockwise turns
  Set Motor [M4] 's Speed to 60 (0~255) Rotating Clockwise turns
  Set Motor [M5] 's Speed to 60 (0~255) Rotating Anti-Clockwise turns
  Set Motor [M6] 's Speed to 60 (0~255) Rotating Anti-Clockwise turns

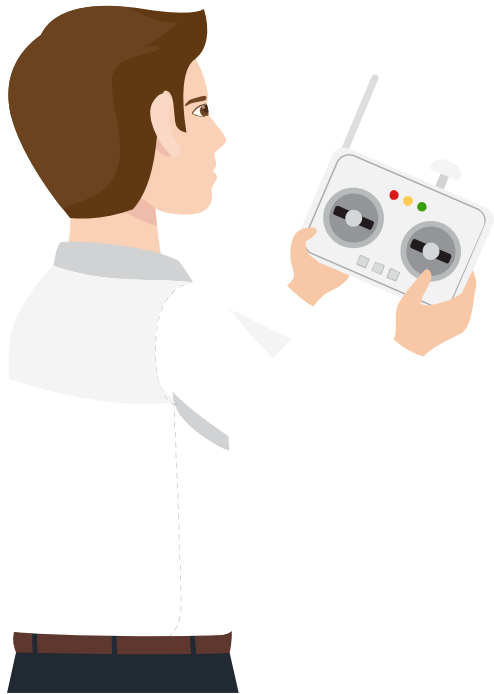
Define [turn_left]
do
  Set Motor [M3] 's Speed to 60 (0~255) Rotating Anti-Clockwise turns
  Set Motor [M4] 's Speed to 60 (0~255) Rotating Anti-Clockwise turns
  Set Motor [M5] 's Speed to 60 (0~255) Rotating Clockwise turns
  Set Motor [M6] 's Speed to 60 (0~255) Rotating Clockwise turns

Define [shoot_ball]
do
  Set Motor [M2] 's Speed to 250 (0~255) Rotating Clockwise turns
  Set Servo on GPIO # [P3] Rotate to 180 Degree (0~180°)
  Wait 1000 Milliseconds
  Set Servo on GPIO # [P3] Rotate to 0 Degree (0~180°)
  Wait 500 Milliseconds
  Set Motor [M2] 's Speed to 0 (0~255) Rotating Clockwise turns

Define [shoot_ball2]
do
  Set Motor [M2] 's Speed to 170 (0~255) Rotating Clockwise turns
  Set Servo on GPIO # [P3] Rotate to 180 Degree (0~180°)
  Wait 1000 Milliseconds
  Set Servo on GPIO # [P3] Rotate to 0 Degree (0~180°)
  Wait 500 Milliseconds
  Set Motor [M2] 's Speed to 0 (0~255) Rotating Clockwise turns

Define [stop]
do
  Set Motor [M3] 's Speed to 0 (0~255) Rotating Clockwise turns
  Set Motor [M4] 's Speed to 0 (0~255) Rotating Clockwise turns
  Set Motor [M5] 's Speed to 0 (0~255) Rotating Clockwise turns
  Set Motor [M6] 's Speed to 0 (0~255) Rotating Clockwise turns
  
```

Projection optimization



Test your car!

P O W E

See you in
the next session!

T H A N K S

P J U S T L E A V E P R E S E N T A T I O N T O O R I

