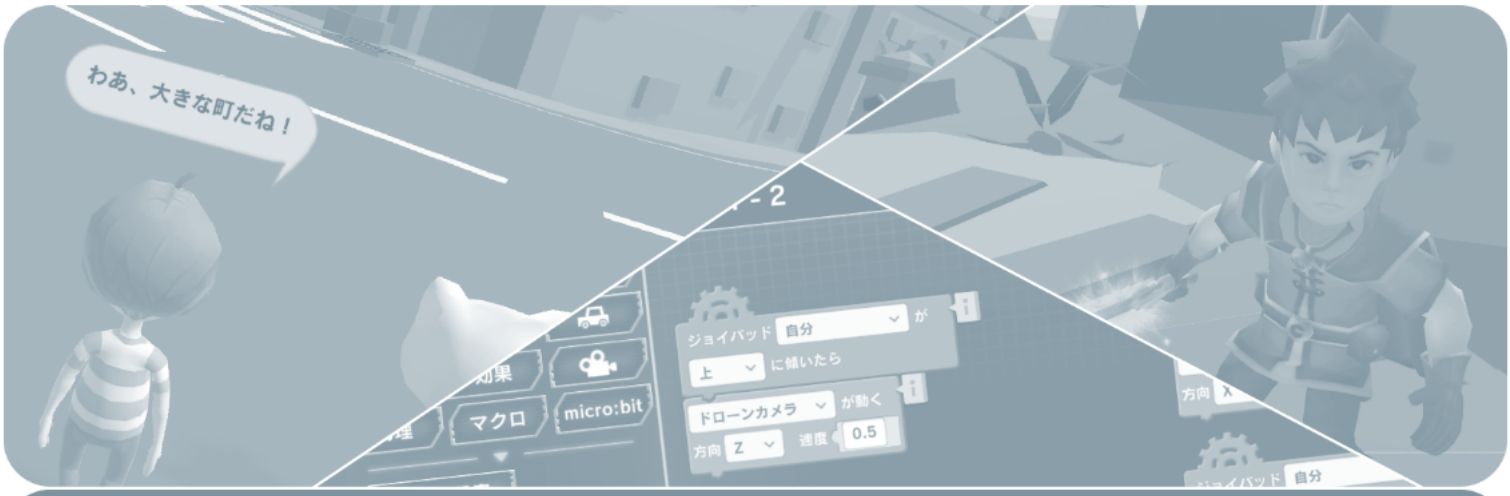



# Mind Render 2

## すぐできる！使い方マニュアル

# Table of Contents

# 1. Features of Mind Render

Mind Render is a programming platform that allows anyone to easily create full-fledged 3D programs. You can create games, simulations, and VR content just by moving and assembling instruction blocks on your computer or tablet. Mind Render was developed by Mobile Internet Technology, Co., Ltd. in 2018 and is available as a free download software for schools and individuals around the world to learn and create programs.

Since its first release, Mind Render has been used in programming and science and math education at national universities, renowned online colleges, and prestigious private middle and high schools. Based on the useful tips for improvements from these educational institutions, Mind Render has undergone a major update in December 2020 as Mind Render 2. Mind Render 2 is easier to use and more feature-rich, allowing both novices and advanced learners in creating and learning programs.

Mind Render 2 has the following excellent features

- Intuitive and easy to understand screen design and user interface.
- Beautiful 3D graphics
- Comes with content that allows you to change the movement of characters without programming
- Programming is very easy. You can create a program by simply assembling command blocks using the mouse.
- You can create a fun 3D world by selecting and combining various components such as characters, spaceships, castles, and skyscrapers.
- Like a movie director, you can freely change viewpoint and direct impressive scenes in any way you like.
- You can reproduce the movement of objects according to the laws of physics to a limited extent without difficult programming (physics engine).
- You can save your own program work in Cloud and have your friends play it (Cloud Locker)
- Compatible with Mac, Windows, iOS, and Android

Despite these unique features, Mind Render is designed to be very easy to learn how to use. Mind Render has 7 rooms of 3 types which are called labs. In each laboratory, you can program people and vehicles in various backgrounds and conduct interesting experiments based on your own ideas. Below are seven labs.

The first type of lab is a "sample program" for those who are new to programming, those who have programming experience but are new to using Mind Render. It explains the basic usage of Mind Render and how to move characters in the 3D world.

In the second type of lab, various programs have been created in advance to provide hints for creating works. You can naturally learn how to program by trying things out in the lab and making changes yourself little by little.

- Let’s make a game
- Let’s make a movie
- Let’s make a story
- Let’s study a little
- Connect with external

The third type of lab is called "Playground", where you can create your own backgrounds and characters without any restrictions. This is a great way to expand your ideas and have fun creating your own programming projects.

**Reference information :**

The "Let’s study a little" lab is detailed in the textbook for Mind Render (2018 edition). In Mind Render 2, there was a change to the command block, so it's not the same, but it's a good reference for understanding the program's concept and overall structure. Please refer to it. The textbook you are reading now introduces other ways to have fun with the same labs, so please enjoy it as well.

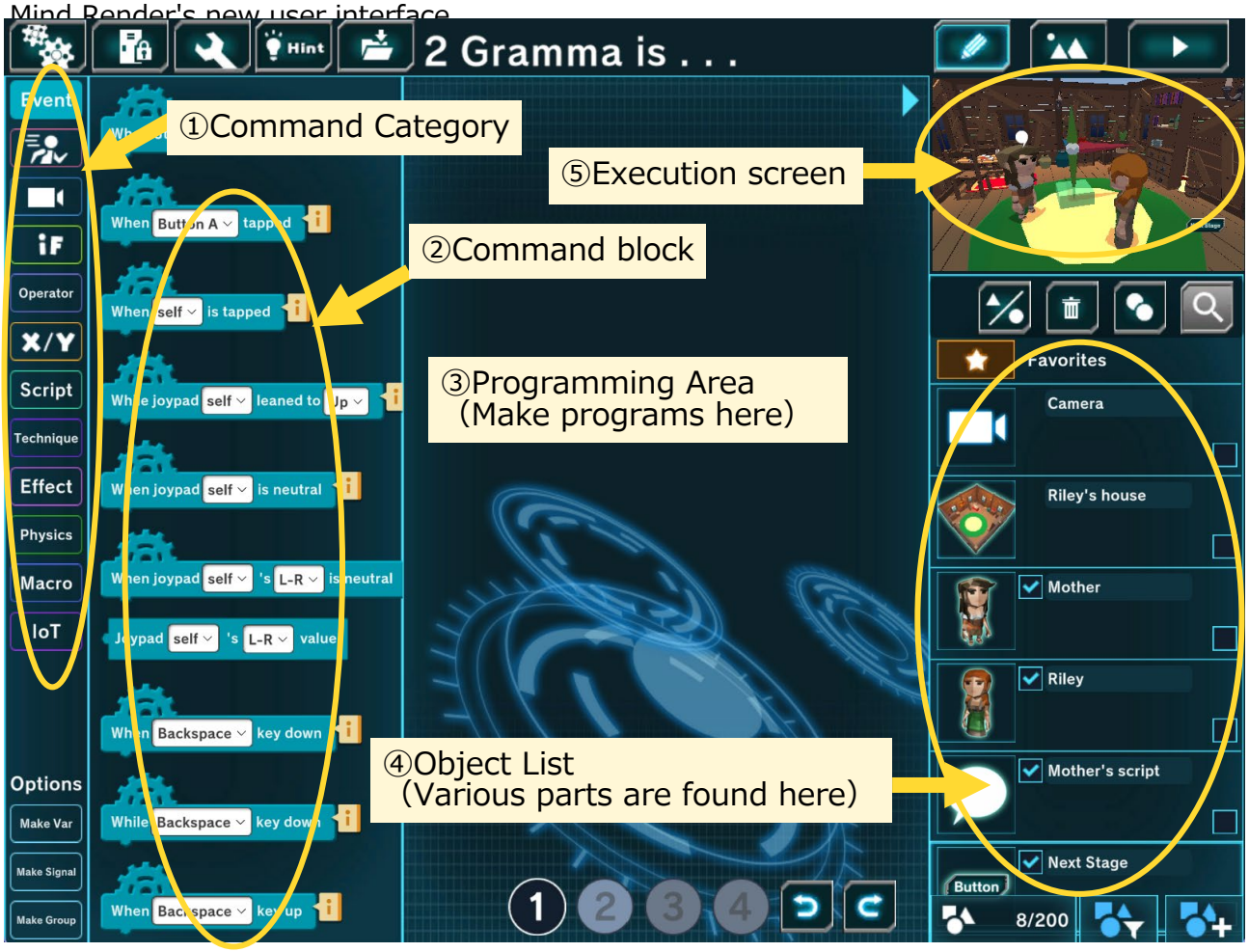
"Nurturing a Genius Programming Workbook on Mind Render", Ryoichi Shirato and others, published by Cutt System (ISBN-10: 4877834362, ISBN-13: 978-4877834364)

# 2. Major update points of Mind Render2

Mind Render 2 has undergone the following major updates with version upgrades.

- The operation screen has been updated for easier and clearer look and use.
- Many new characters and backgrounds have been added.
- Easier to create different types of content such as games and stories.
- Menu categories are arranged in one place so that you can quickly find the command block you want to use.
- Command blocks have been reviewed and consolidated to make them easier to understand and use.

Mind Render's new user interface



# 3. Things you can do with Mind Render2 !

## If you are new to programming

Using three labs for beginners, you can learn the basics of programming while having fun as if you were playing a game. A sample program, Forest Girl, and Space Boy are the beginner's labs.

We will start with a very easy challenge, so even those who have never programmed before can do it with confidence.

**Forest Girl**

This is a story-based content like the Little Red Riding Hood. In this story, a rabbit gives you hints to solve problems and your。

**Space Boy**

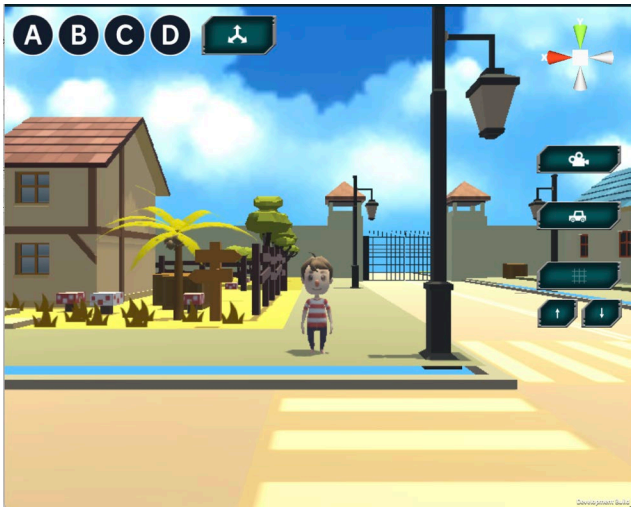
Take a boy and a cat from outer space into the city and help them challenge games against the pirates. Create programs to solve math problems, shoot balls to defeat pirates, and more!

## If you have played Scratch before

If you know the basics of block programming, you can customize content of the labs to learn the basic of Mind Render programming。

Read Chapter 6 “Let’s move the drone” for fun drone programming challenges! !

## If you are teaching programming



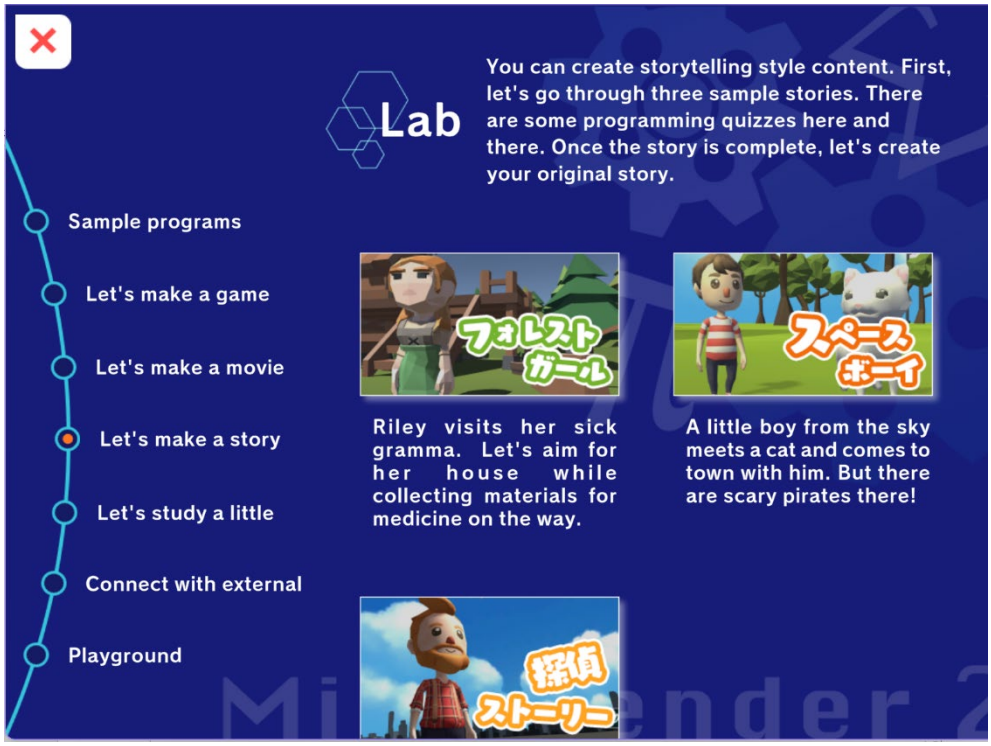

All of Mind Render's contents can be freely modified with your ideas and ingenuity. You can also use it as a teaching material for teaching basic programming skills and basic knowledge of science and mathematics at schools or cram schools.

# 4. Let’s make a story 1 : Forest Girl

## Forest Girl

When you launch Mind Render 2, click " What do you want to create?".


Select “Forest Girl”.



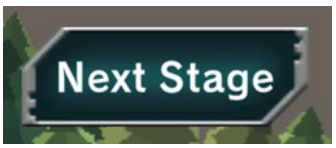
Each lab has two buttons, “New” and “Continue”. If you are using the lab for the first time, select " New", and if you want to continue from where you left off last time, select "Continue". Here, select "New" for "1 Bird’s-eye view of the village ".


## 1 Full view of the village

The figure below shows the startup screen of " Full view of the village ".

Then tap the play button below to start the Forest Girl story.

As you can see in the figure below, explanations are displayed along with the progress of the story, so you don't have to worry about what to do. The first stage is the lab (1-1). Depart from the starting point and head to the goal point, Gramma's house, while accomplishing errands.

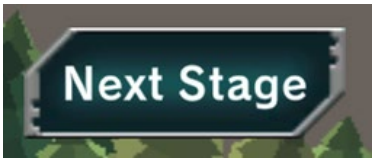

Once the animation explains the story, tap the "Next Stage" button below to continue.

## Advanced Challenge

From the time you tap the “Start” button until the "Next Stage" is tapped, you will be able to see the entire view of this village from above. You can watch this as many times as you like by taping the “Start” button. If there is anything in the village that interests you, you can watch this scene repeatedly.

# 2 Gramma is ...

The next stage is "2 Granmma is...". The explanation will follow soon after the program starts.

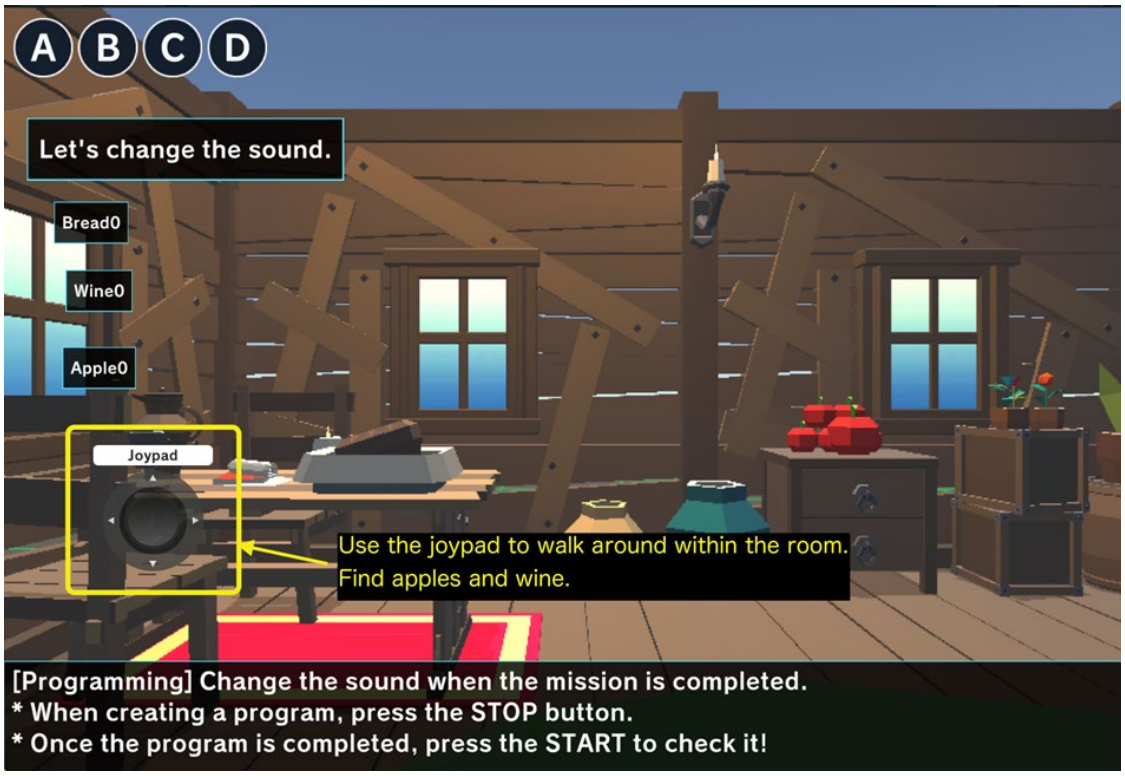

When the explanation finished, click "Next Stage" to continue.

## Advanced Challenge

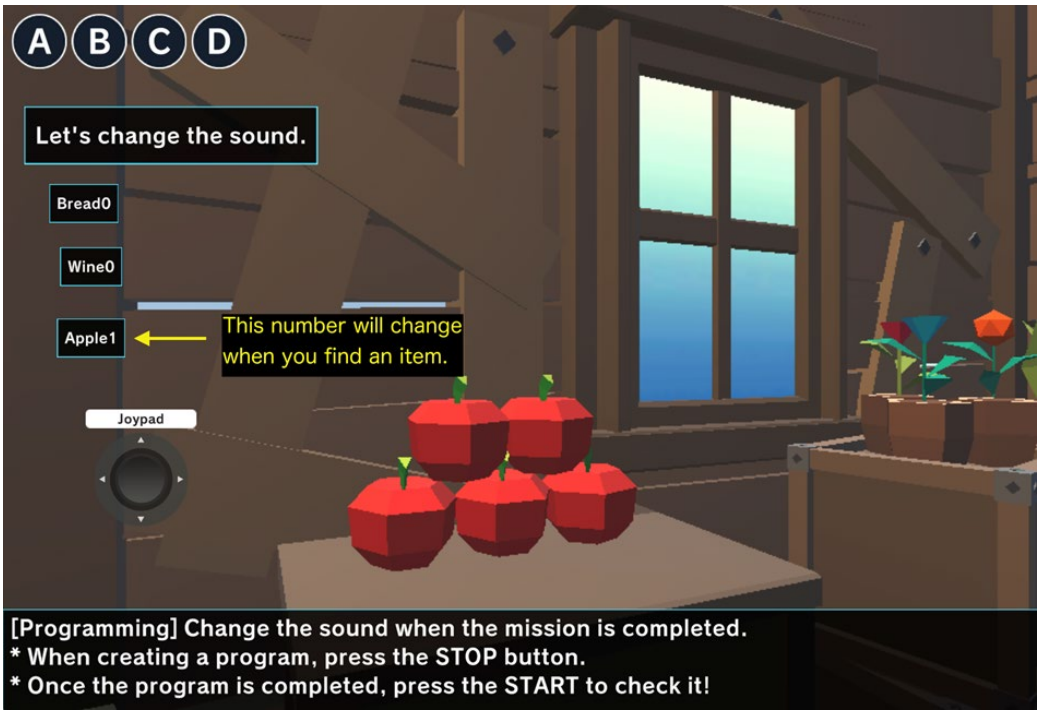

When the program is not running, you can use the trackpad or other controls on your computer to zoom in for a close-up view of the room or zoom out to see the room from the outside, as shown on the left. Give it a try。

# 3 Get-well gift

This is stage “3 Get-well Gift”. In this stage, you will try programming for the first time. Here, programming means looking at the already existed program, finding out what kind of movement it is and making some minor changes to change the sound to be played when the mission is completed. Here the mission is to find and get all three items in the room: bread, wine, and apples. When you complete this mission, you will hear a sound. Your challenge is to change that sound. First, let's see if we can clear the mission by executing this stage.
Use the joypad shown above to move around the room and search for bread and other items.



When you find all three items - bread, wine, and apple - a sound is heard. The challenge here is to try to change the sound. The following below explains how to modify the program to change the sound.

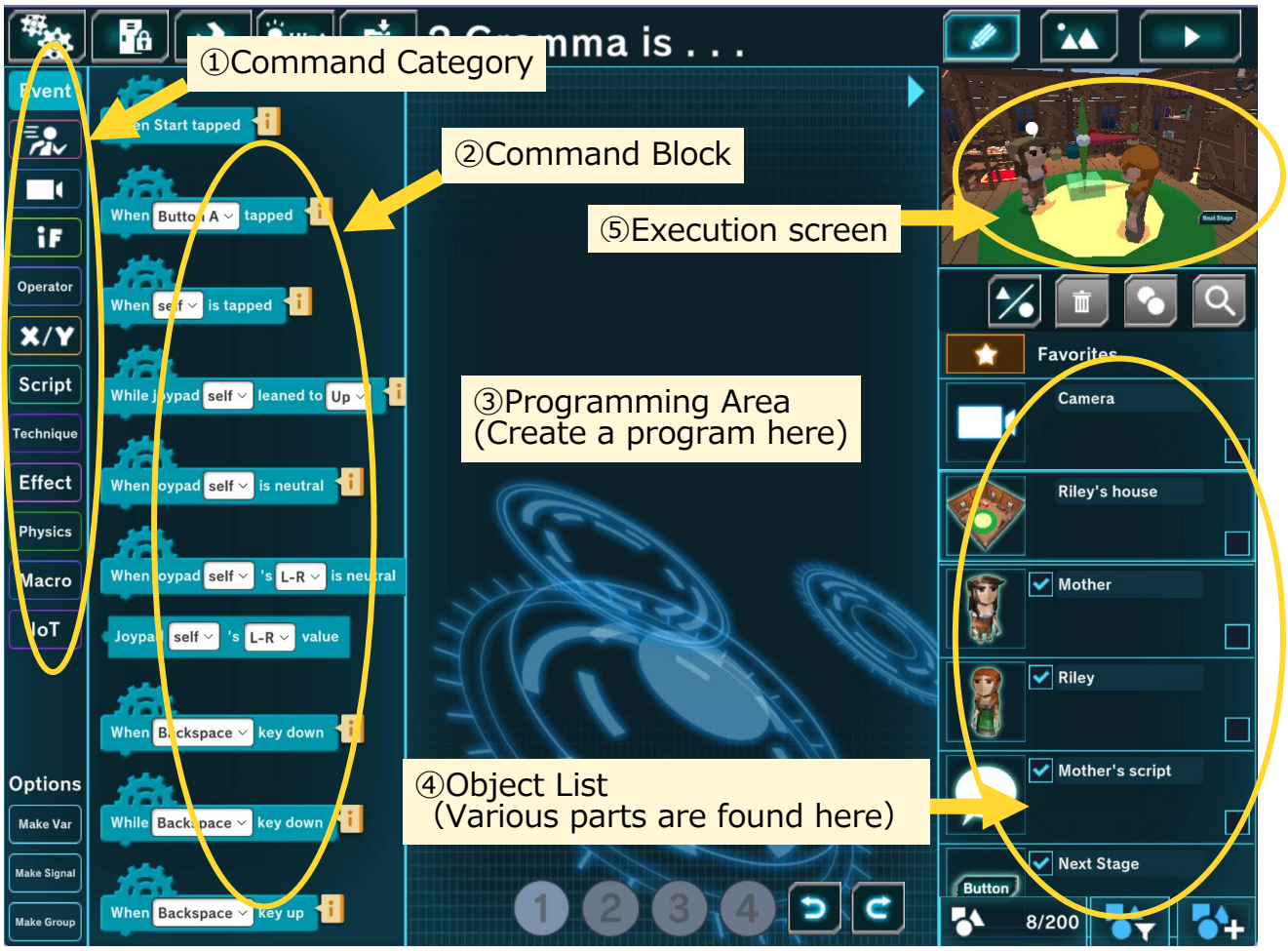



1） Click the STOP button to stop the program from running.。

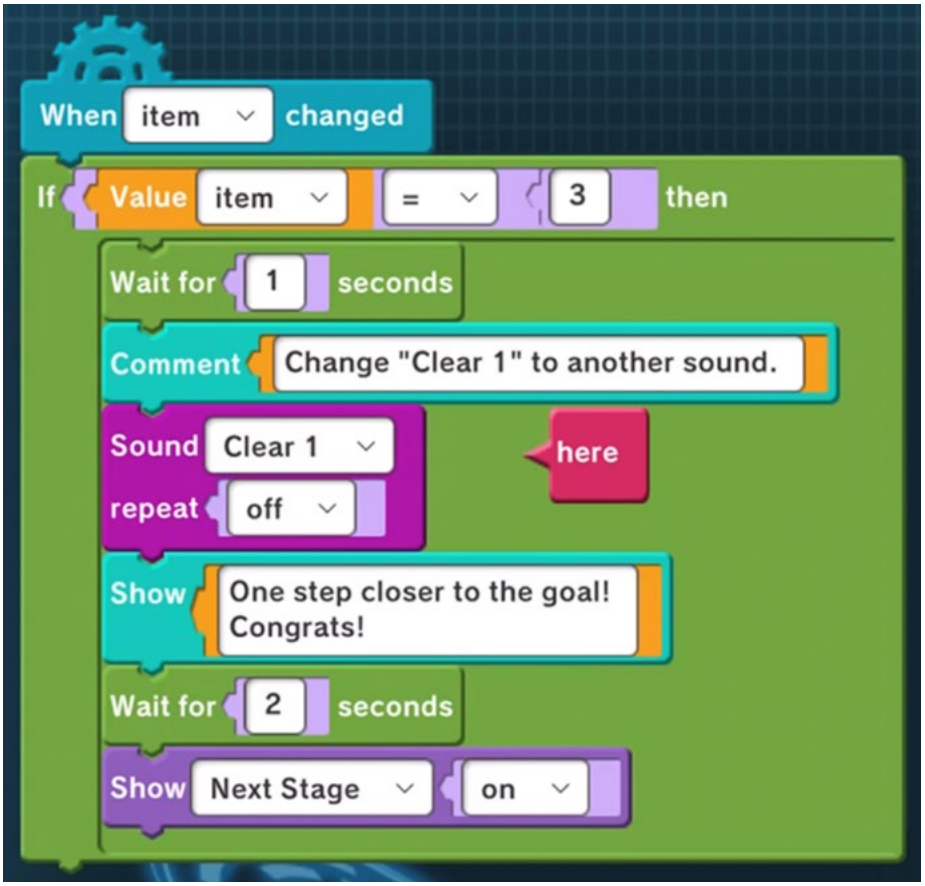
When the program is running, the STOP button looks like the figure on the left. Tap the button to stop the program.

When the program is stopped, the START button looks like the figure on the left. Tap the button to execute the program.

2) Click the pencil button to display the screen as shown below.




3) In the above figure, find the object "Program" in the ④ Object List and click on it. Then, the program will appear in the ③ Programming Area as shown in the figure below. In the center of the figure below, there is a red block labeled "Here", and the brown command block that is labeled "Sound (clear sound 1)". If the text is hard to read, you can move the red "Here" block.




4) The challenge in this stage is to change the sound. If you click on the (Clear 1), you will see choice of other sounds, as shown below.

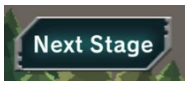

If you want to listen to the sound first before choosing from one of the three clear sounds, select one in the object list and tap on it. You can hear what it sounds like.


5) To select a sound, tap on "Program" from the object list to display the program once more.

6) Click the Start button to run the program to see if it sounds the way you want it.

If the sound for each item (bread, wine, and apple) is what you specified, you have succeeded.

If you are successful, tap the "Next Stage" button to proceed to the next stage.




### Advanced Challenge

If you decrease the number of “Wait ～ seconds”, the waiting time will become shorter. If you increase the number, the waiting time will become longer. Try making the number smaller or larger.

## 4 Let’s go

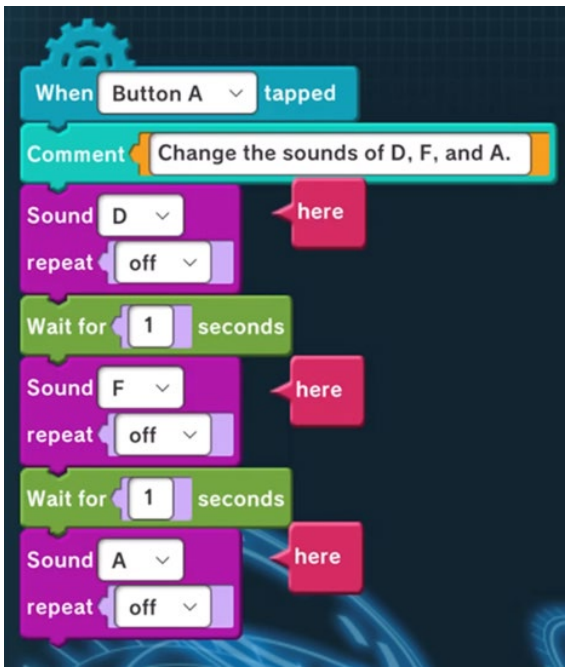
In the next stage “4 Let’s go” you will meet a rabbit. Once you meet the rabbit, tap "Next Stage" to go to stage “5 Meet a rabbit”.

# 5 Meet a rabbit

In stages “5 Meet a rabbit”, you will be challenged by a rabbit. The challenge from the rabbit is to make the same sound that the rabbit makes.

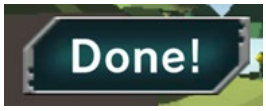
Tap the Start button to stop the program running.

Click the pencil button to display the programming screen.



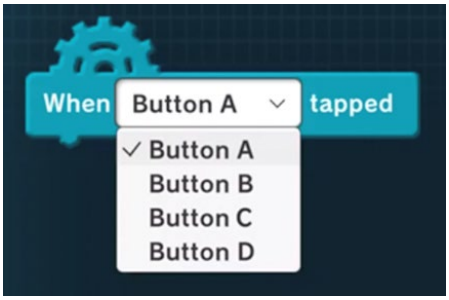

When the screen changes, you will see the program as shown below.

Tap "D", "F", and "A" in the program to select from other notes. Choose the notes and their order so they sound the same as the three notes the rabbit plays in the previous stage “5 Meet a rabbit.”



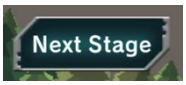

If it works, tap the “Done!" button on the left to proceed.

## Advanced Challenge




In this program, a sound is played when Button A is tapped. You can change Button A to Button B or C. Click Button A as shown on the left to bring up the pull-down menu and select a button you want.

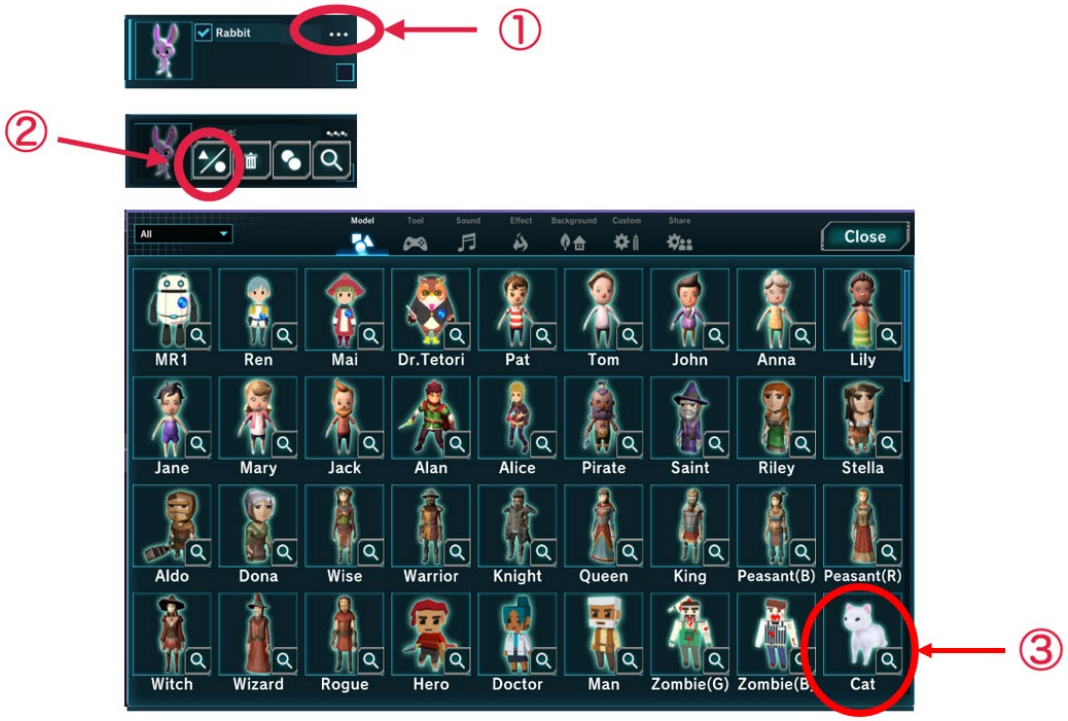

The rabbit will tell you what you need to do for the challenges ahead. You need three colors of medicinal herbs (red, blue, and yellow) and a golden egg to make a medicine for grandma.




Click on the "Next Stage" button to proceed to “6 Collect herbs.”

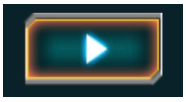
## Advanced Challenge

You can change the rabbit to another character. Switch to the program editor and follow the order of the numbers in the figure below to change the rabbit into a cat.



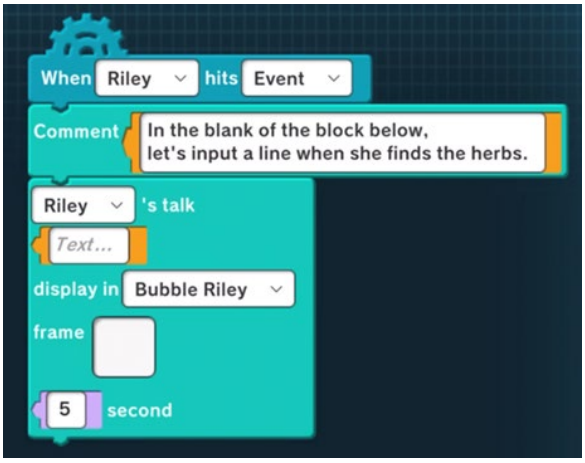

## 6 Collect herbs

In the stage "6 Collect herbs", you will find three colors of medicinal herbs (red, blue, yellow). You have one more programming challenge. The challenge is to "Enter the lines for when you find the medicinal herbs ".

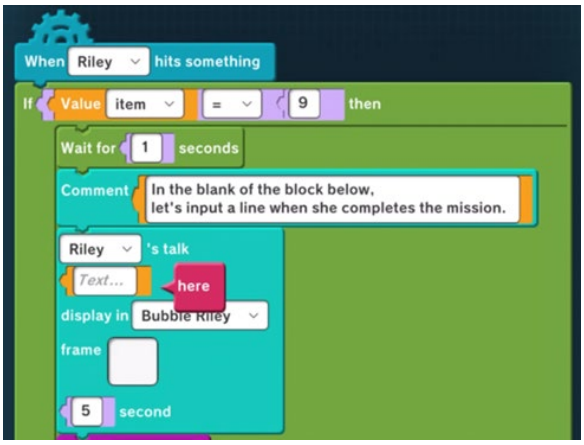


Tap on the STOP button to stop the program from running.

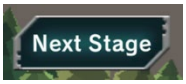
Click the pencil button to display the programming screen.




In the first program, just below the "Riley's talk", put your favorite line in the place marked "Text...". For example, if you enter "I found it!”, the first time she finds a herb, she will say, "I found it!”.




In the second program, enter a line to be displayed when all the medicinal herbs have been found in the required number and the mission has been cleared. For example, "Yay! I completed the mission!” etc.



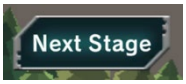

If you are successful, tap the "Next Stage" button and proceed to “7 Wolf.”

## Advanced Challenge

You can select a color by clicking the "Background" square in the dialogue instructions. Let's change it to your favorite color.

# 7 Wolf!

In stage “7 Wolf!”, a wolf will attack you. As you run away from the wolf, you will meet a hunter who will help you defeat the wolf.


When you meet the hunter, tap the "Next Stage" button to continue.

## Advanced Challenge


It is difficult to tell how far away the wolf is in the above figure. When the program is not running, you can switch to a horizontal view by tapping the icon pointed by the red arrow. To switch back, tap the camera button under the 3D icon circled in red. Let's give it a try.

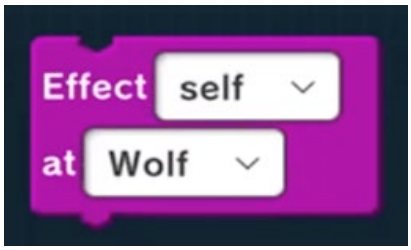
# 8 Help me, Hunter!

In stage “8 Help me, Hunter!”, you will be challenged to program using sounds and effects 。 The content of the challenge is " When a hunter and a wolf come into contact, make sounds and effects!".


Let's display the program in the figure below by referring to the explanations in Stages “3 Get-well gift” and “6 Collect herbs.”

It will be good if you change the three places shown by "Here". There are hints in the "Comment" about what sounds to make at each of the three places.

The hint says, "Play Attack sound.", so you know you should do as shown in the figure below.

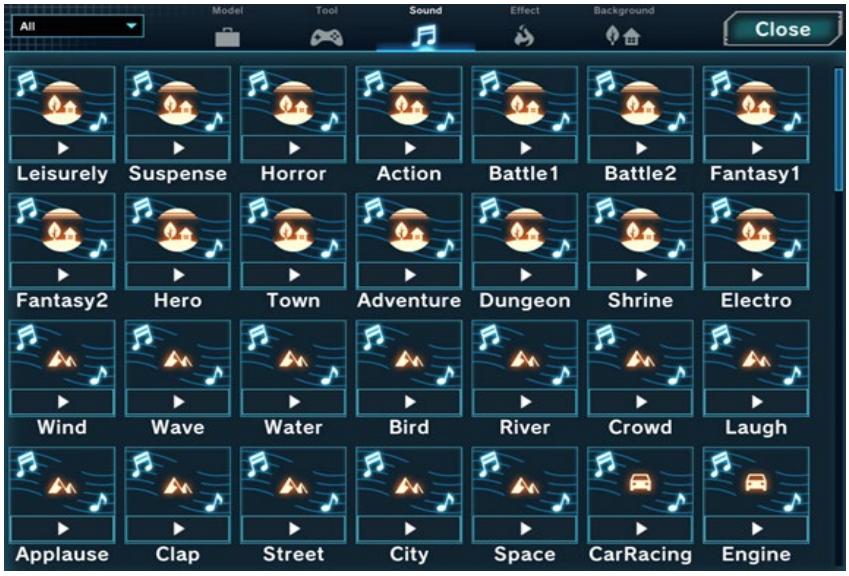

Note: In the effect block, you need to specify not only what effect you want to show, but also "where" you want the effect to appear. In this case, you want the wolf to disappear along with the smoke, so the position of the effect should be at wolf.

Once you have completed the program correctly, experiment with different sounds and effects that you like.


If it doesn't work, tap "Try again". The message "Check the program!” will be displayed. Find out what you need to fix in your program.

When you are satisfied with the result, tap "Done!”. Click on the "Next Stage" button to proceed to the next stage.

### Advanced Challenge

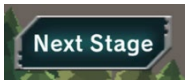



You can freely change the sound to be played. Open the object menu in the program editor, select your favorite sound, and play it.

## 9 Something from the sky

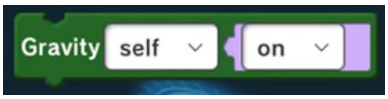
As the stage "9 Something from the sky" begins, something suddenly falls from the sky and cries out, "Help!”. When you use the joypad to get Riley there, you will see it is Saint. Saint is hungry and wants you to share some food with him.


Click "Next Stage" to proceed to the next stage “10 Reward for helping”.




## Advanced Challenge

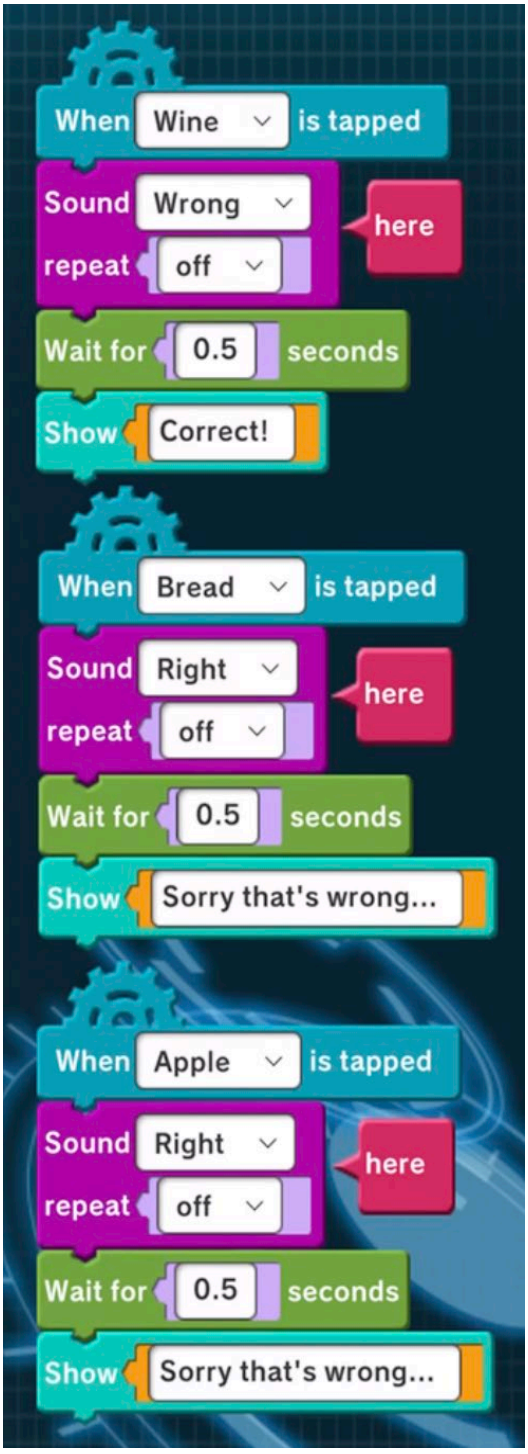
Let's look at the program for stage “9 Something from the sky” in the program editor. In the Saint's program, there is a "Gravity (self) (on)". This is to enable the effect of gravity. Then Saint will stay in place when he falls from above. Let's change the option (on) in this command to (off). Saint will then fall from above and when he hits the ground, he will bounce back and fly away. Try it yourself.



# 10 Reward for helping

It seems that Saint wants either an apple, bread, or wine, so tap the three buttons to find out which one he wants.

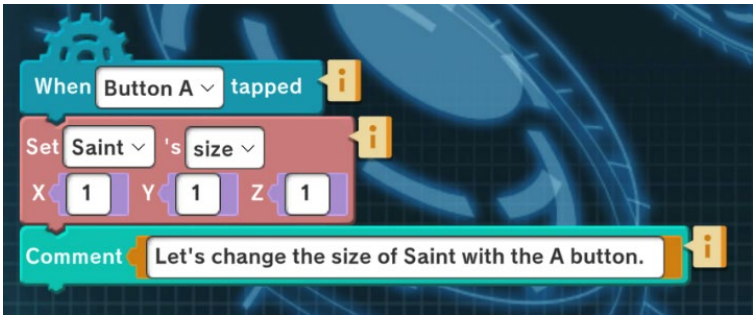
Once you know what Saint wants, it's time to try your hand at programming. Let's display the program shown in the figure below, referring to the explanations in stages “3 Get-well gift” and “6 Collect herbs.”



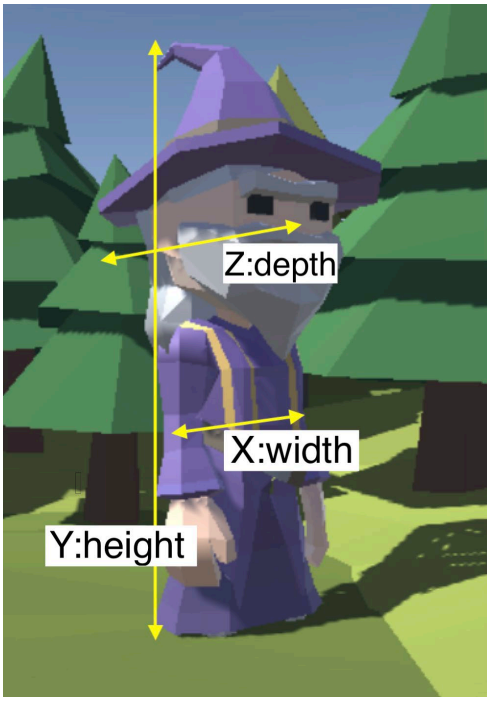

If you look at this program through, you can tell wine is the correct answer.

The program on the left shows the "Wrong" sound when wine is selected, even though it should be the "Right" sound. Also, when you select bread or apples, the program should make the "Wrong" sound, but these two wrong answers will give the "Right" sound.

Change the sound of the wine, bread, and apple so they will give you the correct sound respectively when tapped.




Once you have changed the sound correctly, tap A button to change the size of Saint. You can change the size of any object, not just Saint, by changing the program.



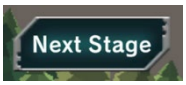

This command allows you specify how much larger in:

X: width

Y: height

Z: depth

For an example, if you want to make it taller, set (Y: height) to a larger number.



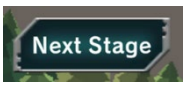

At the end of this stage, Saint will give you a chicken that lays golden eggs. After you get the chicken, tap the "Next Stage”, move on to stage “A golden egg.”

### Advanced Challenge

Let's make Saint fatter or much smaller by referring to the explanation of specifying the size of Saint above.

## 11 A golden egg

In this stage, the chicken lays a golden egg.




Once you have the golden egg, tap "Next Stage" to move on to stage “12 Gramma’s house.”

### Advanced Challenge

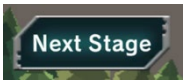
Modify the program so a sound is played when the "Next Stage" button is tapped.

## 12 Gramma’s house

In stage “12 Gramma’s house”, you will finally go to the Gramma's house. Use the joypad to get to Gramma’s house.

### Advanced Challenge

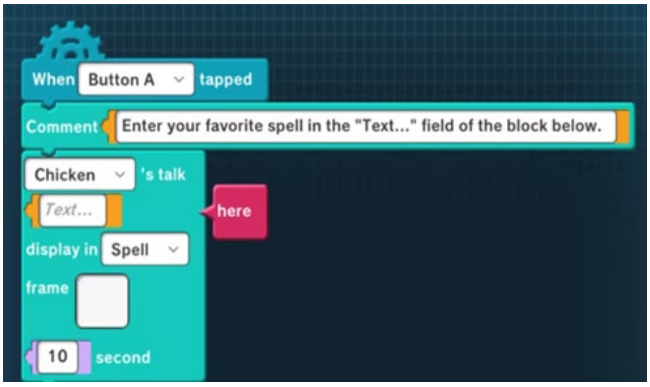

You can change Riley’s walking speed faster or slower by changing the speed in the command shown on the left. Try it.



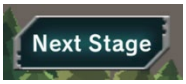

When Riley arrives at Gramma's house, tap the "Next Stage" to move on to stage “13 Make medicine.”

# 13 Make medicine

In stage “13 Make medicine”, your challenge is to make the chicken to say a spell. Bring up the program shown below, referring to the explanations in stages “3 Get-well gift” and “6 Collect herbs.”


In the program shown left, type in your favorite spell in the "Text..." field under the "Chicken's talk”. For example, if you type “Bibbidi-Bobbidi-Boo”, the chicken will say this spell when you tap the A button.


Once you have the medicine, tap the "Next Stage" to proceed to the last stage “14 The end.”

### Advanced Challenge

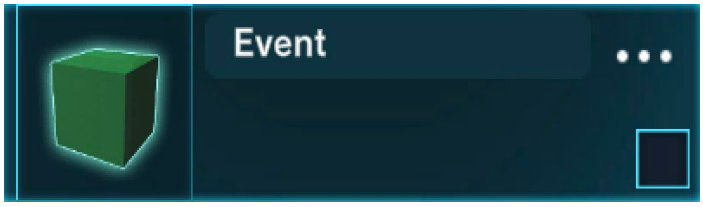
Play effects when the chicken casts a spell to make them more effective. Try different effects to see which effect is best suited.

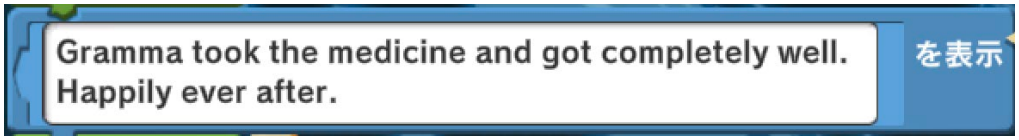
## 14 The end

In stage “14 The end”, Gramma got well because of the medicine. This is the end of the Forest Girl's story. What do you think? With just a few tweaks to the programs, you can change the story in any way you like.


## Advanced Challenge

To change the text appeared at the bottom of the screen in the above figure, change the program of the "Event" object in the object list. Have some fun to change the text.



# 5. Let’s make a story 2: Space Boy

## Space Boy

“Space Boy” is a lab with a storyline, just like “Forest Girl”'. Follow the instructions on the screen and in the program.

At some points, you will be given program assignments. Some of them are a little more complicated compared with the assignments in "Forest Girl", but there are plenty of hints. Command blocks and sample programs are provided for your reference.


Select "Space Boy" from the lab menu.

## 1 Something has fallen?

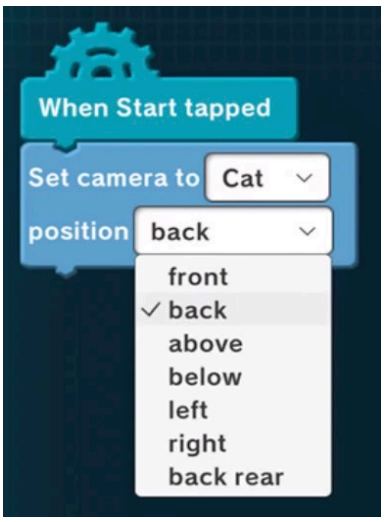
The beginning of the story.

Tap the "Start" button to see the story.



When the animation is finished, tap the “Next Stage" button to proceed.

## Advanced Challenge



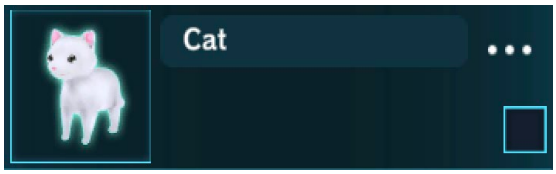

Look at the animation from a different view. Inside the camera object, make the pull-down menu visible as shown in the left figure. Try changing the camera's mounting position to see how the view will be different.

## 2 Let’s go see

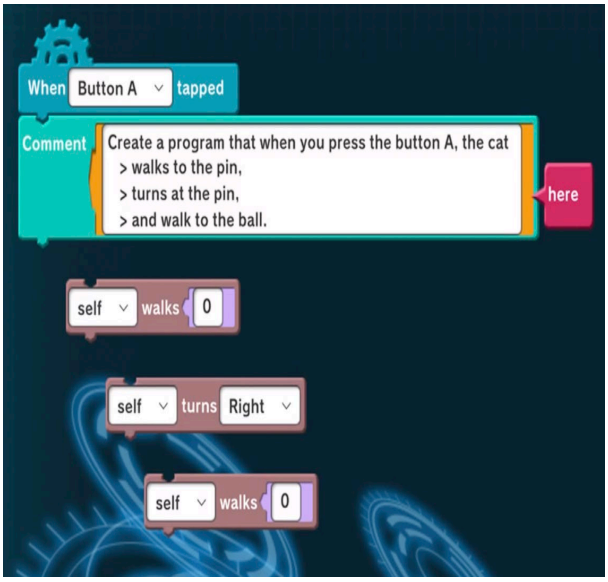
At this stage, you will challenge programming. Create a program to make the cat walks to the dropped ball. The command blocks needed have already been placed, so combine them be careful of the order.


When creating a program, click the button on the left to display the programming screen.

The same is true when creating programs in subsequent stages.




Since you will be creating a cat program, scroll down the object list to find "Cat". Tap the picture of the cat then you will see blue frame around.




The "Cat" program will appear in the programming area. Scroll down to the bottom and you will see a red block saying "Here".

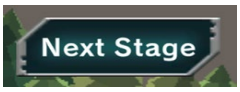
Use the comments to help you create the program. The comments do not affect the behavior of the program, so you can keep the comments. You can connect the brown blocks.

For the number of steps, put a number between 10 and 20. Also, choose the direction to turn.

Once the program is ready, run it to see how it works.
Tap the Start button to run the program.


You made the cat walks when button A is tapped, so tap button A.

When the cat gets to the right position, a small boy will appear from the ball. If the cat goes too far or goes over, adjust the number of steps.




If you are successful, tap the “Next Stage" button to continue.

## Advanced Challenge

Let's look at the place from above to see where the cat and the pin are。
Go to the scene edit screen and click the cat image in the Object List.
The screen will change, and it will change to the scenery seen from the back of the cat.。

**For Windows**

Change angle: move the mouse while taping the right mouse button
Move: Hold down the trackball and move the mouse
Change Height: Rotate Trackball

**For macOS**

Change angle: Hold down the option button on keyboard and move three fingers on the trackpad

Move: Hold down the control button on keyboard and move three fingers on the trackpad.

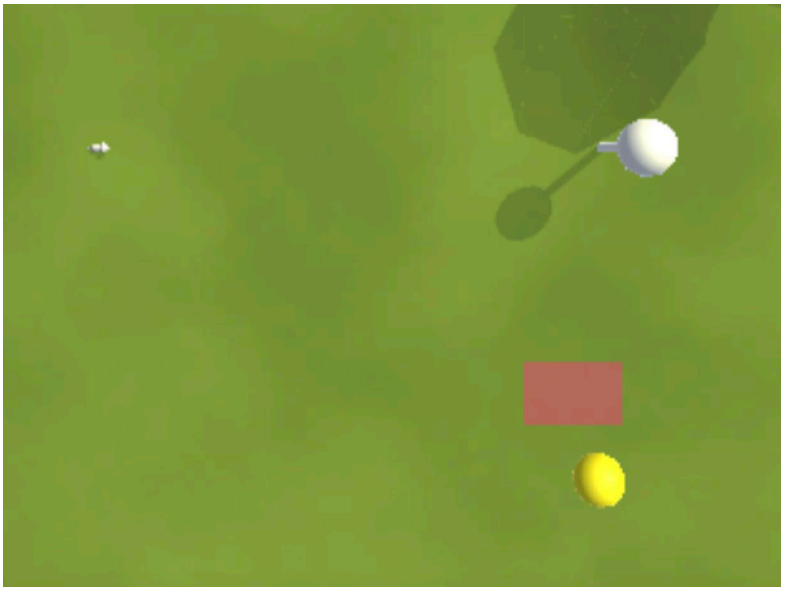
Change height: move two fingers on the trackpad

**For touch screen**

Change the angle: move two fingers

Move: Move 3 fingers

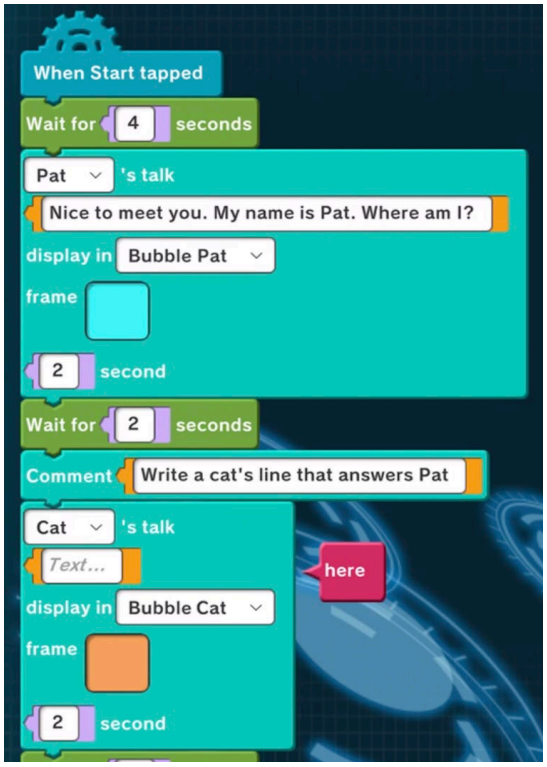
Change height: pinch in/out



You can clearly see the positional relationship between the cat, the pin, and the ball.

## 3 Nice to meet you

The cat and the boy exchange greetings. Only the boy's script is included in the program. Add the cat's script to complete the program so they talk to each other.


Make the program in the “Cat" object.

Scroll down to the bottom of the cat's program and you will see a red block saying "Here".

Read the boy's script and think of script for the cat. Tap on the blank space marked "Text..." in a light color. Then you will be able to enter scripts.

There are two places for the cat's scripts.
Make sure to enter scripts in both places.

When you complete your program, watch how their conversation goes. Tap the START button.

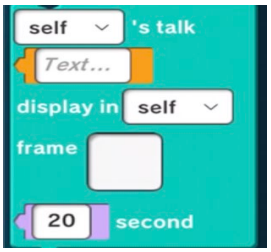

If you are successful, tap the “Next Stage" button to proceed.

### Advanced Challenge

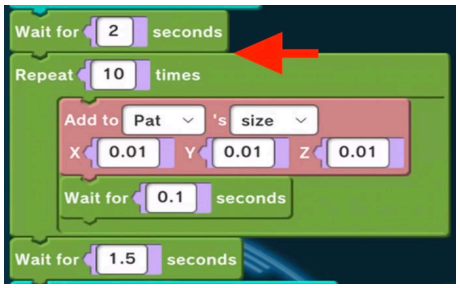
Let’s try to rewrite Pat's script.

You can add commands to make their speech longer. The procedure for adding commands is as follows.

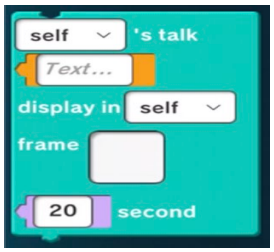

1）In the command category, tap "Script".




2）Scroll down to the bottom of command, find and drag the command shown below into the programming area.




3）Insert the command into the program. For example, insert it before the repeat command.

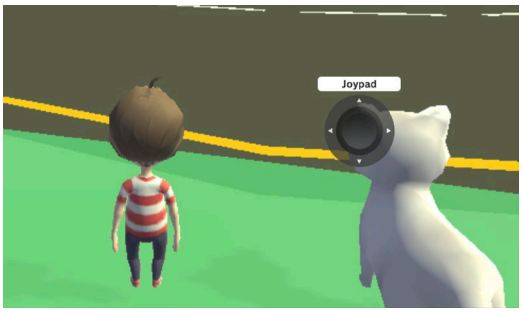



4）Enter the script.

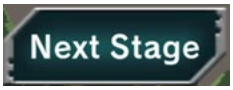
Specify whose scrip it is, how many seconds it is displayed, etc., referring to other script commands.

## 4 Let’s go to the town

Two of them are going to the town. Go to the town by using the joypad. The bridge in front of them is the entrance to the town.


Go to the green area past the bridge.




Once you have arrived, tap the “Next Stage" button to proceed.

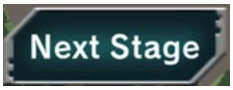
**Advanced Challenge**

The entrance to the town can be found somewhere else. Walk along the wall and you will find a gap where the boy and the cat can walk through.


They can enter the town through the gap and walk around the town. However, you cannot proceed to the next stage because the “Next Stage" button is not displayed. To proceed to the next stage, redo the lab “4 Let’s go to the town” and enter the town from the proper entrance.

## 5 Pirates are here!

When they arrived at the town, some pirates came over. What will happen to Pat and the cat? Let's see.


When the animation is finished, tap the “Next Step" button to proceed.

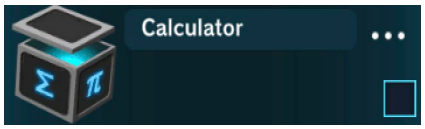
### Advanced Challenge

The pirates that appear in this lab are created by changing the size of the same pirate character in the X, Y, and Z directions to create pirates with different sizes and aspect ratios. The left figure is the program in each pirate object.

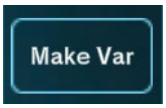
Let's change the size of X, Y, Z in this and change it to a giant pirate, a small pirate, etc.


## 6 Oh no!

To solve the pirate’s spell casted on Pat and the cat, you have to solve a programming challenge. In this stage, you will see a sample program. In the next stage, you will create your own program.

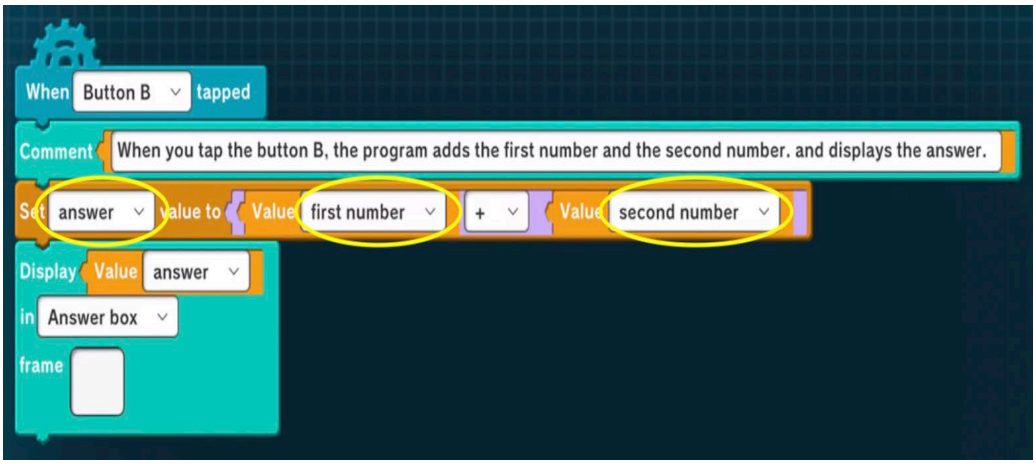




The programming challenge is to make a program that adds two randomly chosen 2 digits numbers.
Make a program in the “Calculator" object.

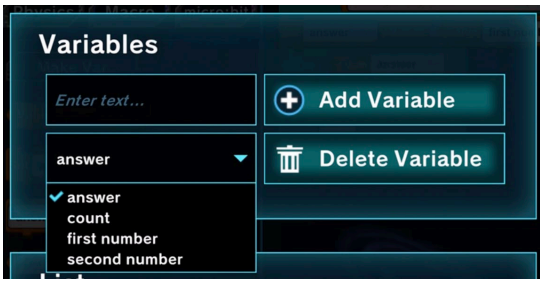



Inside “Calculator", you will find the following program.

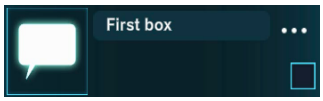
"answer", "first number", and "second number" are variables. A variable is a convenient mechanism that allows you to change its contents or add or subtract numbers from its contents. You can create your own variable and name it anyway you like.

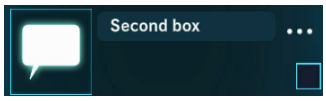



To create a new variable, click "Variable" at the bottom of the command category。



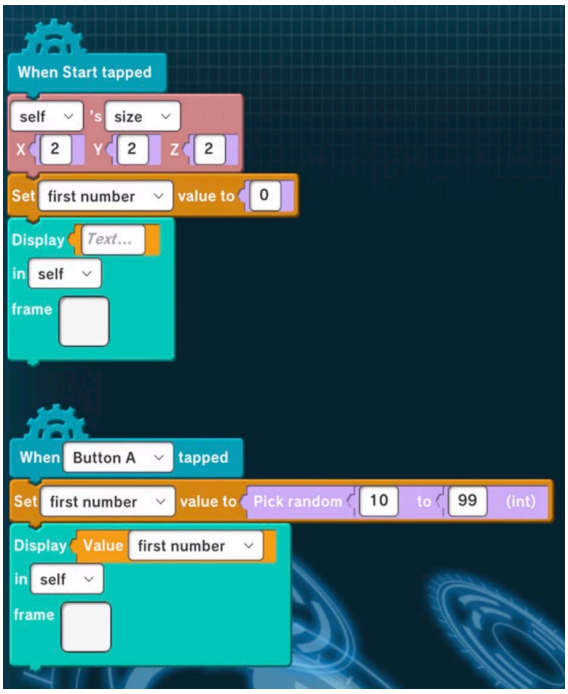

In this stage, the variables have already been created. Here, click the "Close" button in the upper right corner to close the screen.




From the above program, we know that the variable "answer" is the sum of the first number and the second number.

Then what numbers are needed for the variables "first number" and "second number"? You can check them in the "First box" and the "Second box" object.

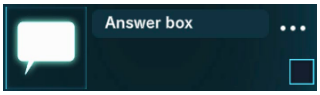



The "First box" is for displaying the first number. The "First box” has the program as shown in the left figure.

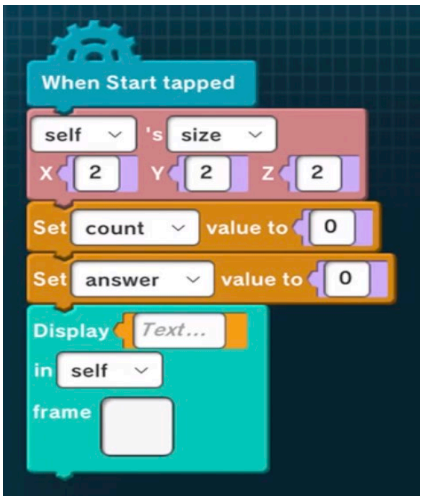
When the START button is tapped, the first number is set to "0". It also clears the old number displayed in the box.

When button A is tapped, the first number is set to one of the numbers from 10 to 99. It will be displayed in the box.

The program for the "Second box" is the same as the “First box”.

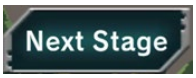



Let's also look at the "Answer box" program.




When the START button is tapped, the number of “answer” is set to "0".

It will also clear the number displayed in the box.



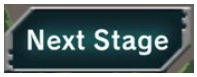

When you check the program and saw how it runs, tap the “Next Step" button to proceed.

## 7 Let’s break the magic

Referring to the previous stage ”6 Oh no!”, make a program that adds two random numbers. The program will be created in the “Calculator" object.

If you want to check the previous stage again, tap "Select Lab" button to select “6 Oh no!”.

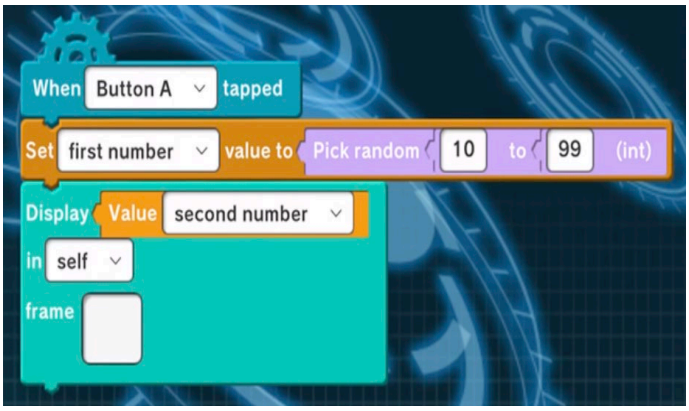
At this time, tap the "Save" button to save the program that is being created so that it will not be lost.


When you have successfully programmed it, tap the “Next Stage" button to proceed.

## Advanced Challenge

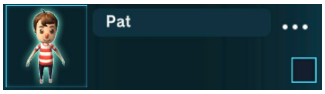
In Lab “6 Oh no!” and “7 Let’s break the magic”, two different 2 digits numbers were displayed each time you tap the A button. In order to create and display a different number each time like this, we use a command called random number. It is a command that determines a range, such as 10 to 99, and then selects a number from that range just like a roll of the dice.



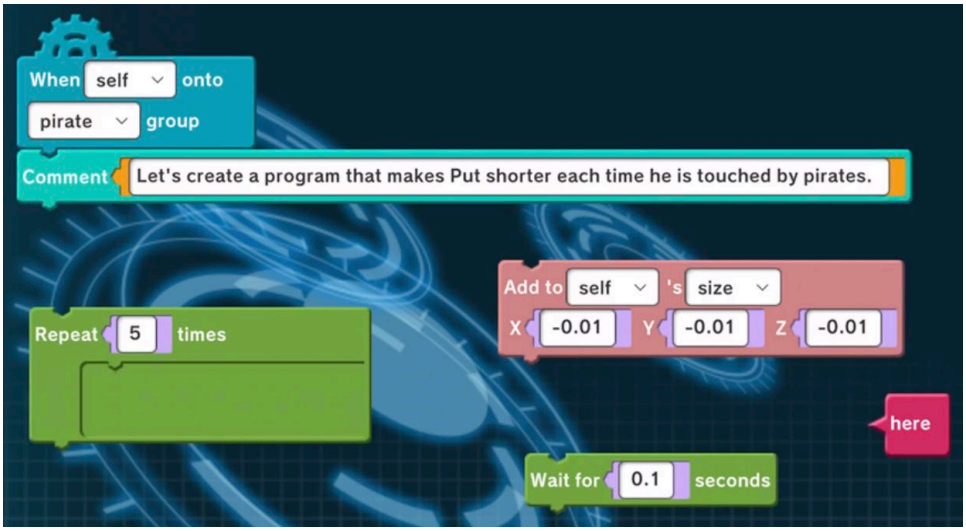

For example, the left figure is a program that creates and displays the first number of two numbers for addition. Among them, "random number (10) to (99)" is an instruction to generate a random number. This command selects one number from the range 10 to 99. You can change the range to generate random numbers by changing the numbers in the command. How can I do calculations with 3 digits? Please try it.


## 8 Run!

The magic has been solved. Pat and the cat can move again now. The pirates are chasing them, so hurry up and run away. Use the joypad to move Pat and and the cat.

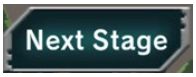




Add an effect that Pat becomes smaller when touched by a pirate. Make the program in the "Pat" object.



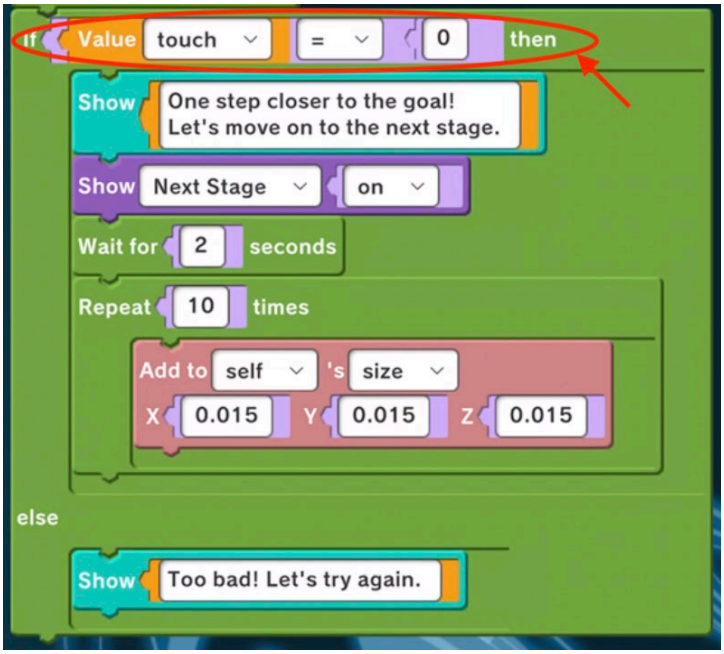

Find the red block labeled “Here".

Connect three command blocks provided so Pat gets smaller and smaller. Try different values like "-0.01" or "5 times" and see what happens.

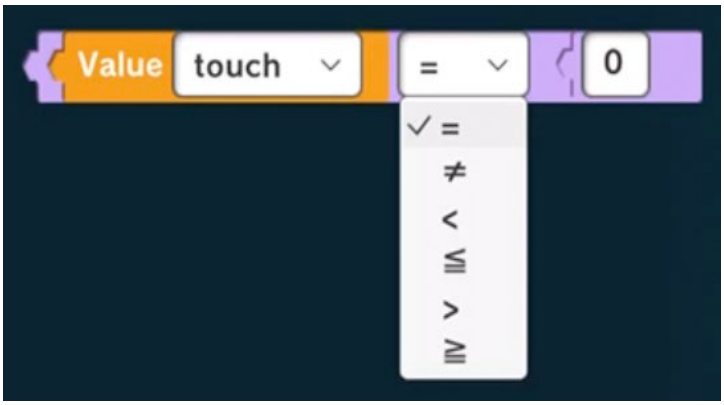



If you are able to escape successfully, the “Next Stage" button will appear. Proceed to the next step.

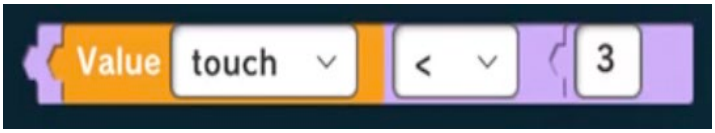
## Advanced Challenge




In order to complete the mission of this lab, you have to avoid being touched by the pirates for 30 seconds. If you feel that this condition is a little too strict, you can change the condition. The program that sets the condition can be found in the "Time left" object in the object list.
The figure on the left shows where the condition is set. In the command in the red circle, the touch value (number of touches) is set to.




Click the equal sign of this command to show other options.

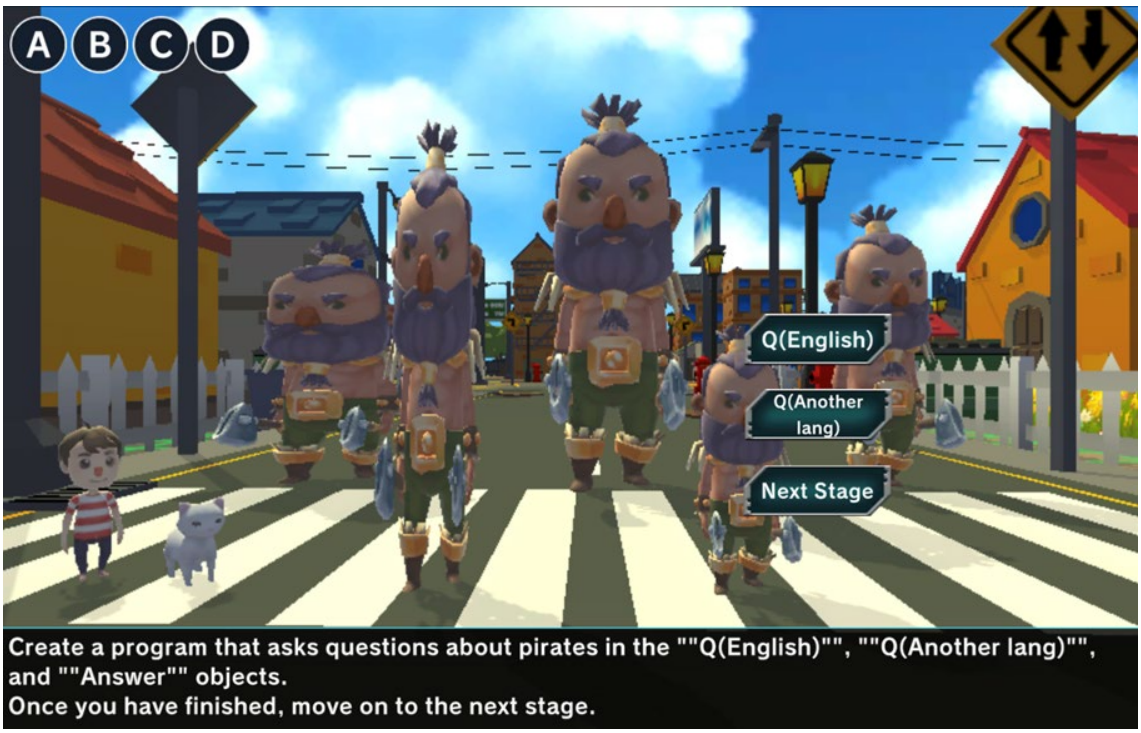


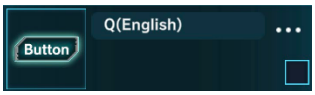

For example, if the touch value is less than 3, you can change the condition to allow up to two touches.

Change this command to set your condition such as "equal to or less than 5" or "less than 3".

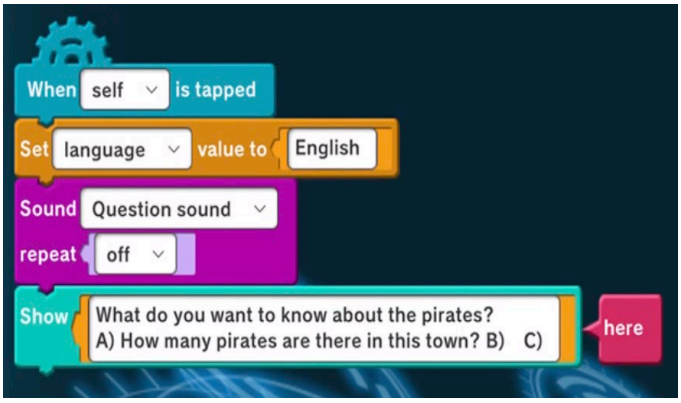
# 9 Get to know the pirates

In this stage, create a program that asks and answers questions about pirates.

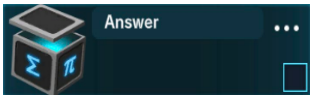


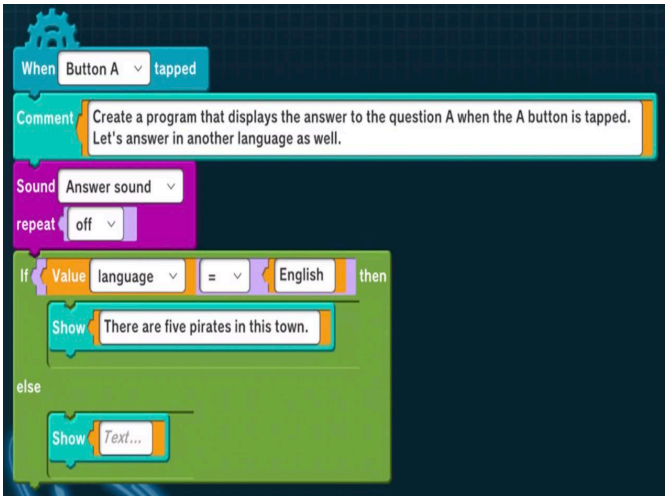

The program that asks the questions is created in "Q (English)” object.




As an example, question A) asks about the number of pirates.
Think of questions B) and C) and enter the texts.
Of course, you can change the content of question A) as well.



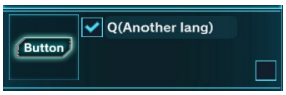

The answer program is created in the “Answer" object.



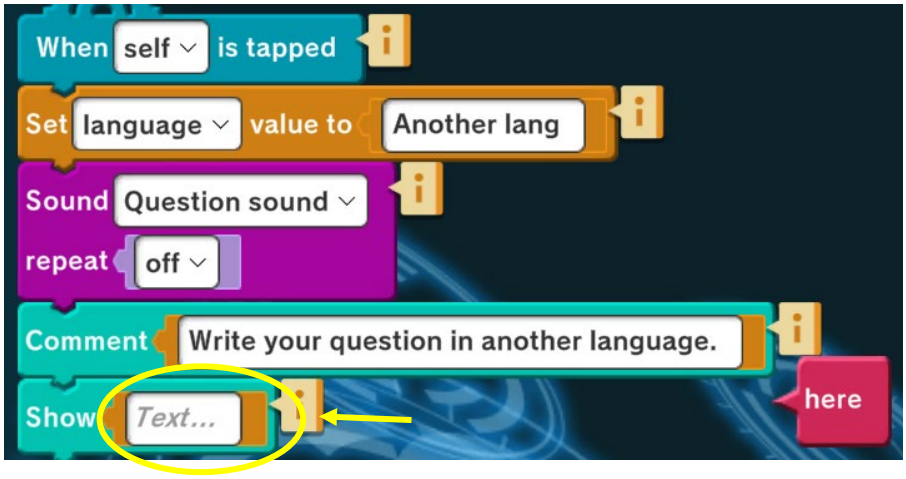

The left figure is an example of a program to display the answer to the question A. When you tap button A, the answer to question A will be displayed.
The content of the answer can be changed freely.
In the same way, make a program that displays the answer to the question B when button B is tapped, and the answer to question C when button C is tapped.

## Advanced Challenge

Let's change the program so that if a question is asked in English, it will be answered in English, and if a question is asked in another lang, it will be answered in the language.



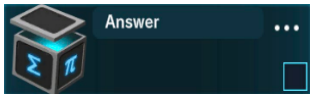

For another language questions, write them in "Q (Another lang)".



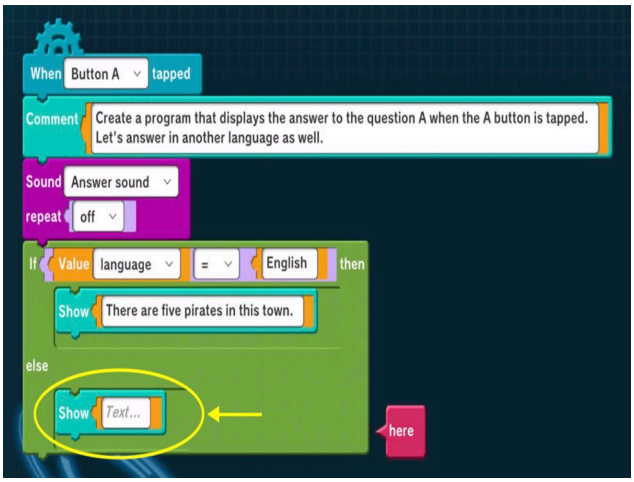

The left figure shows the program in the Q (Another lang) object.

Enter a question in another language in the area labeled "Text...".

When the “Q (Another lang)" button is tapped, the value of the variable "language" is set to “another lang".
This determines whether the answer should be displayed in English or in another language.



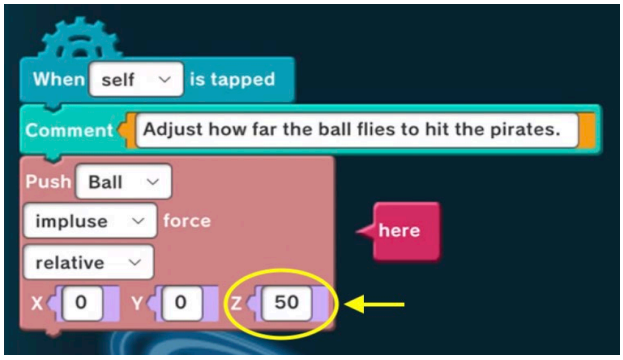

Type in the answer to the question in the object "Answer".




The figure on the left shows the program inside the object "Answer".
Enter the answer to the question in English in the area labeled "Text...".
By the “If” command, if the value of variable "language" is English, the answer will be displayed in English, and if it is not English, the answer will be displayed in the language you entered.

## 10 Beat the pirates

This is the last stage. Here, you have to hit the pirates with the balls to fight back. If you hit all the pirates in time, you will complete the mission.

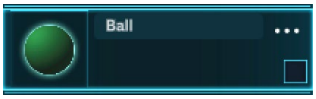



You can adjust how far the ball fly by using the programs in the "Launch1", "Launch2", and "Launch3" in the object list.
The left figure shows the "Launch1" program.

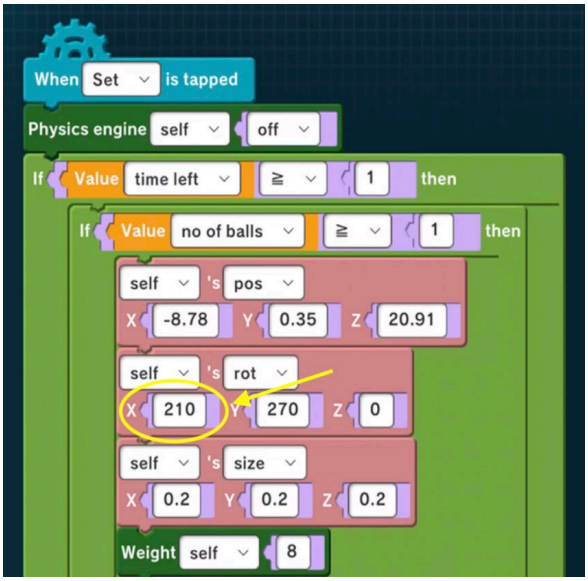
To shoot the ball, push the ball with an impulse force. Impulse force is to push something out instantaneously.
Since we want the ball fly forward, we apply the force from behind the ball. In the command block, this is specified as relative option and the value of Z-axis.

Relative option is a specification method that applies force from the side or back of an object based on the object. The Z axis is the axis that extends in the front-back direction of the object.
How far the ball fly depends on the amount of force you apply.

The same applies to the "Launch2" and "Launch3" programs.




You can also change the angle of the ball.
Let's take a look at the "Ball" program.




The value of angle "X" is set to 240 (degrees).
The value of angle "X" is set to 240 (degrees).
The X axis is the axis extending from left to right, and the angle at which the object will be rotated along the axis is the X angle. For an example, imagine spinning on an iron bar.
Change 240 degrees to 50, 180, 210, etc. to see what the result looks like.

This concludes the story of Space Boy. How did you like it? The programs that make the story are all created with command blocks. Feel free to change the programs yourself to create more interesting adventurous stories.

## Advanced Challenge

Once you are comfortable with programming, take a look at programs beside challenges.

Modify the programs to create an original story. For examples,

- Replace the characters with something else.

  Note: You may need to adjust the size of the characters.
- Replace effects and sounds to whatever you like.
- Feel free to change or recreate the story as you like.

# 6. Let’s make a story 3: "Detective Story"

## Detective Story

Lab “Detective Story" is a lab with storyline, just like "Forest Girl" and "Space Boy".

One day, the detective receives a request letter. The content of the request was "I want you to protect the hidden treasure". Can he protect the treasure? And what is the hidden treasure...?

Let's proceed to solve the mystery.

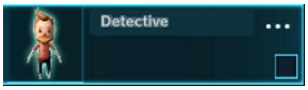
### Characters


Select "Detective Story" from the Lab Experiment menu.

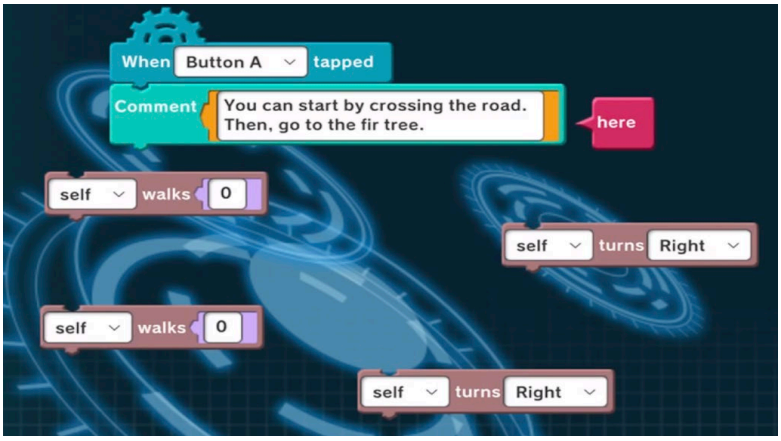
# 1 Letter of request

Tap the "START" button to see the story.

The detective will go to the client Marie's house to hear her problem. Mary's house is marked by a fir tree. Tap the A button and make a program to let the detective walk to the house.

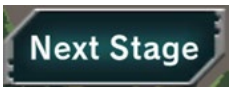




Create the program in the “Detective" object.




Make the program referring to the comments.


The mission is completed once you arrive in front of Marie's house.



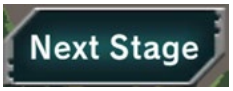

Tap the “Next Stage" button to proceed to the next stage.

## Advanced Challenge

here is more than one way to get to Marie's house. For an example, you can start by moving to the right and then move forward. Create a program that will take the detective to Marie's house in this way.

# 2 Notice

Go inside of Marie's house to hear details of the request.


When the animation is finished, tap the “Next Stage" button to continue.

## Advanced Challenge

How do you get the camera face to the apple while Marie is talking about the treasure?

Hint: The apple is next to Marie (on the left side, relative to the direction Marie is facing). It looks like you should place the camera at Marie's position and point it to the left.

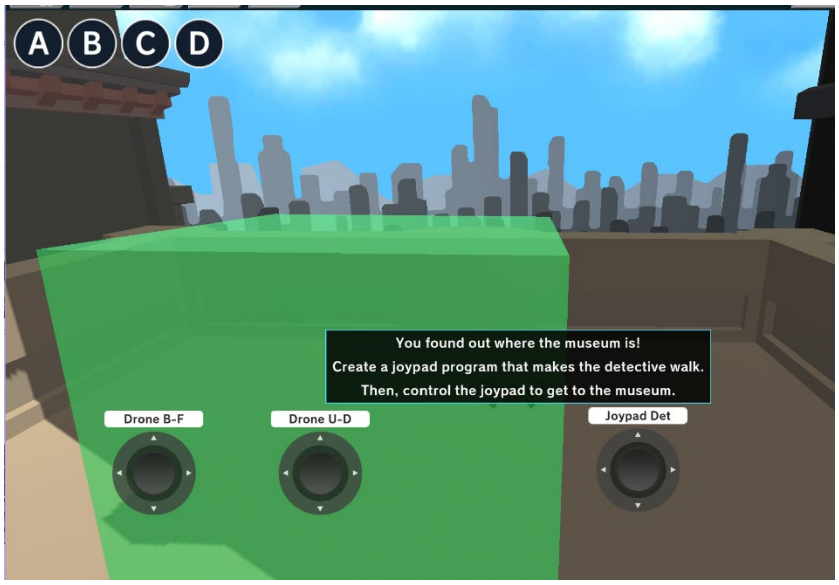
# 3 Go to the museum

I'm going to the museum where the treasure that Marie told me about is on display. There are two challenges in this stage.


First, fly a drone to search the location of the museum.

There is a red drone in the center of the screen.
There are 2 joypads to control the drone. On the roof of the museum building, a green box is placed as a landmark. Use the joypads to control the drone to find the green box.


You could see a green box on the roof of the building.

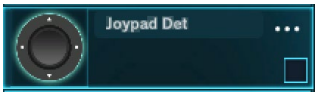



Once the drone touches the green box, the first task is completed.

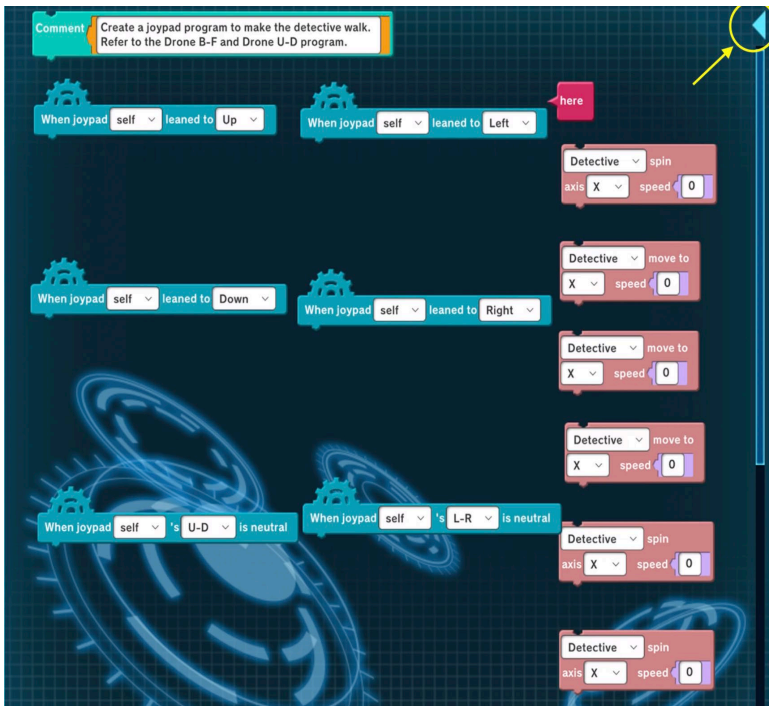
After it touched the box, the following task is displayed.

The next task is to program the joypad to take the detective to the museum.

Display the programming screen.



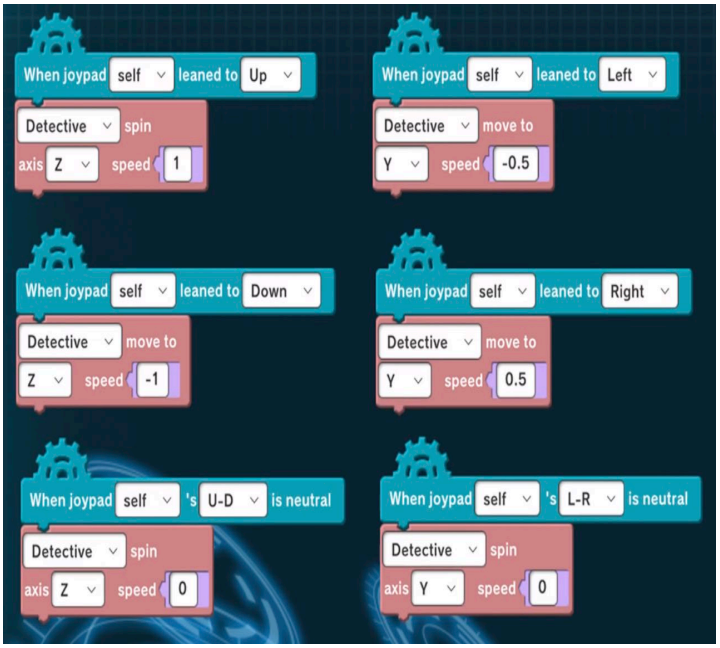

Make the program in the “Joypad Det" object.




Click the arrow to enlarge the screen.

The command blocks you need have already been placed. Make your own program referring to the drone's joypad program.

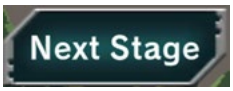
Specify Z for forward/backward movement and Y for left/right rotation.




The figure on the left is an example of a joypad program.


Control the joypads to get to the front of the museum.

In front of the museum, there is a green area, a landmark.




If you are successful, tap the “Next Stage" button to proceed.

## Advanced Challenge

When you walk around in the large city, it is difficult to tell whether you are approaching the landmark or moving away from it. It would be useful to display the distance between the detective and the landmark.


The figure on the left is an example of how to do it. This program always displays the difference between the x-coordinate of the landmark and the x-coordinate of the detective. If you move in the direction where the difference becomes smaller, you can see that the distance to the mark becomes shorter in the X direction.


When the program shown above is running, the distance will be shown as in the left figure. Use the above program as a reference to display the distance in the Z direction as well.

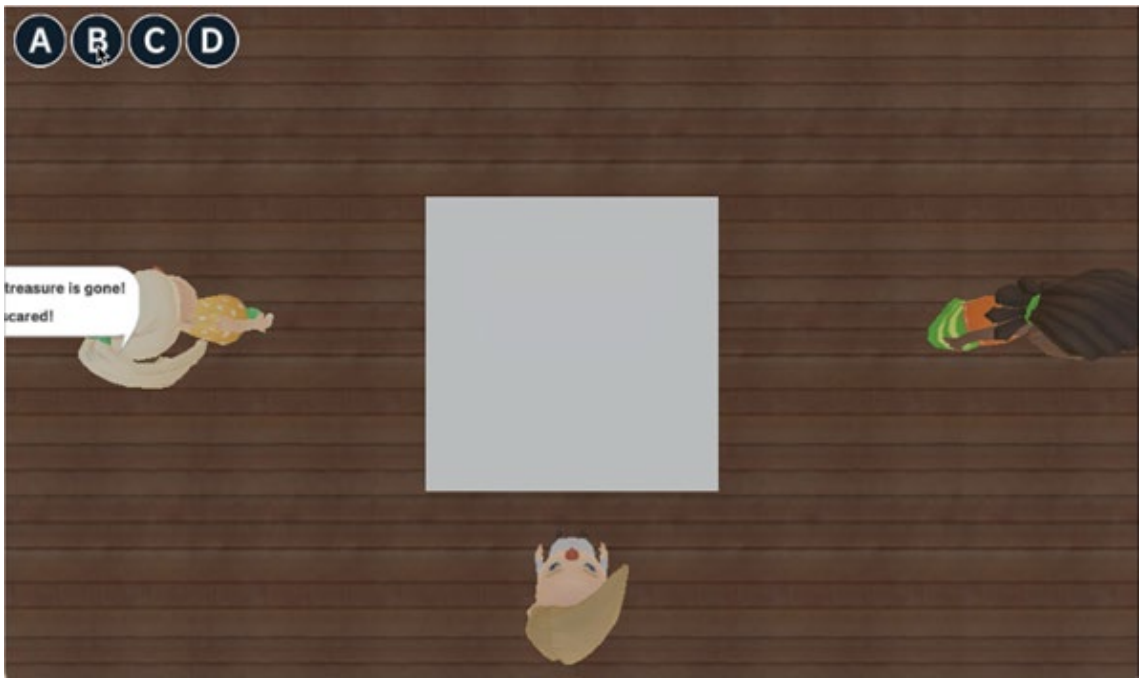
## 4 The hidden treasure is…!

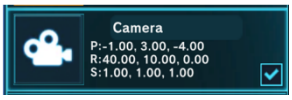
We came to the treasure room in the museum. Let’s talk to the security guard.

However, the hidden treasure disappears right in front of the security guard and the detective. Switch the camera position so that we have a better idea of what exactly happened in the room.

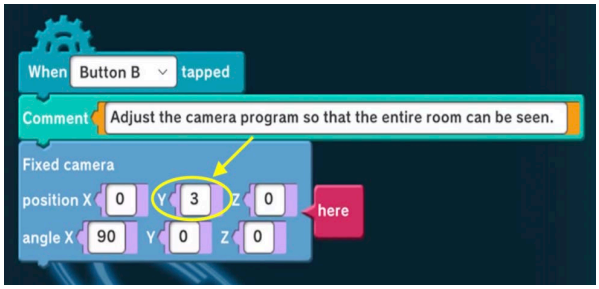

Tap button B to switch the camera position to have a view from the ceiling.

However, only part of the room is shown. Adjust the camera program so that the entire room can be seen.




Displays the “Camera” object’s program.

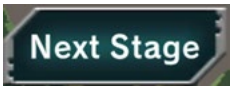



Experiment with where to place the camera to capture the entire room.

For example, try to make it to have a view like this.

To change the height of the camera, change the coordinates of Y-axis, which is the vertical axis.

Try various other values to see how the view changes.

If you are successful, tap the “Next Step" button to proceed.

## Advanced Challenge

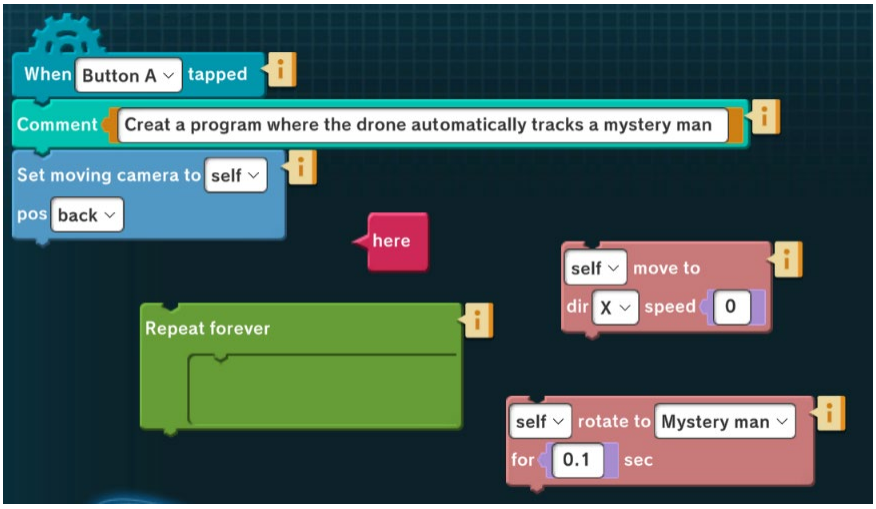
How can we switch camera perspective of the camera to the guard's or detective's point of view using the C and D buttons? Let's make a program to switch the camera view.

# 5 Mystery Man

There was a man running away from the museum. Let's make a drone auto-tracking program and follow the man.



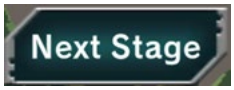
Make the program in the “Drone" object.




The command blocks you need are already placed. You can connect them to make the program.

Watch your speed so that you can track the man a little behind.

If the mystery man makes contact with the mystery woman, make the drone stop as well.


If you are successful, tap the “Next Stage" button to proceed.

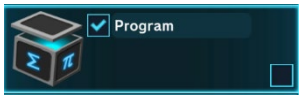
## Advanced Challenge

Referring to the advanced challenge in Lab “3 Go to the museum”, make a program to display the changing value of distance between the fleeing man and the drone.

# 6 Reason for running away

The detective catches up with the man and tries to find out what happened to him. To get the man to talk, the detective has to win rock-paper-scissors.


Let's make a rock-paper-scissors program so the detective always wins if you tap the A button.




Make the program in the " Program” object.

The following programs are on the program’s tab pages 1-4 of the rock-paper-scissors program.

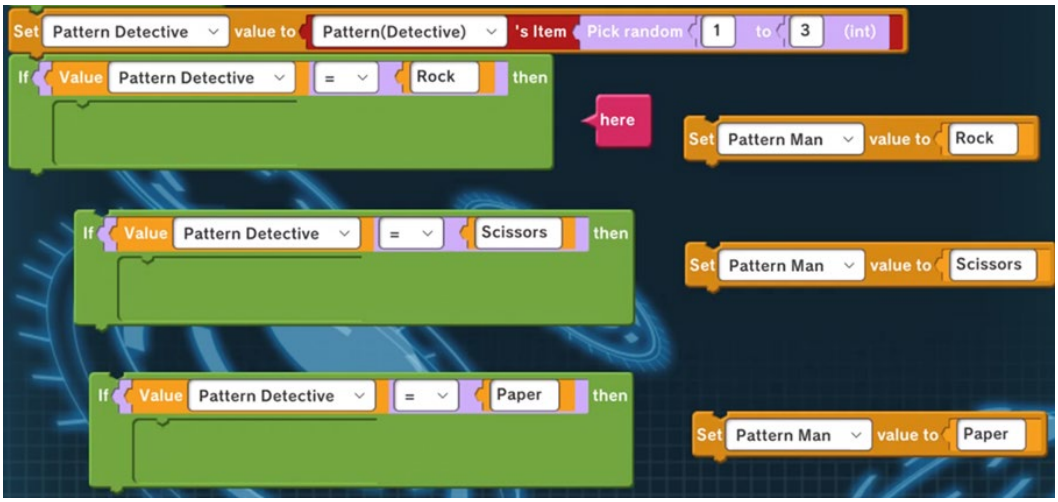
Lists, macros, and programs commonly used in pages 2 to 4
(Do not change 2-4 programs as they may stop working.)

Program where the detective always wins when button A is tapped. (Incomplete)
Complete the program.

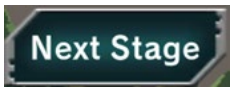
Program that randomly determines the winner when button B is tapped.
(Completed)
Tap the B button to run the program.

Program that determines the winner in three games (Completed)
Tap the C button to run the program.

The figure below is a part of the program on page 2. Connect commands for rock, scissors, paper so that the detective always wins.




Try the programs on page 3 (B button) and page 4 (C button) as well



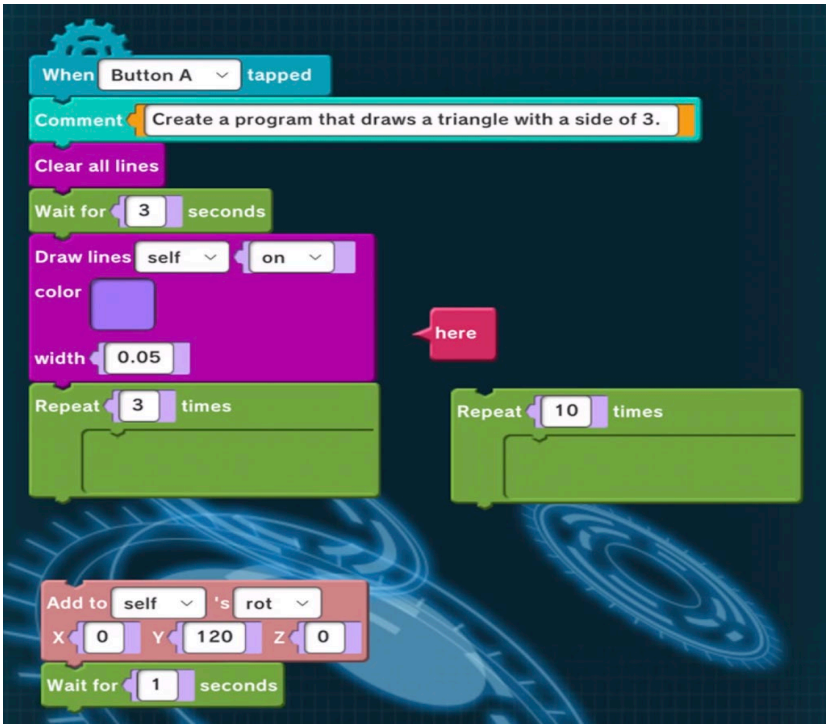

If you are successful, tap the “Next Stage" button to proceed.。

## Advanced Challenge

Let's make a program that the detective always loses when the D button is tapped, referring to the program when the A button is tapped in the "Rock-paper-scissors program".

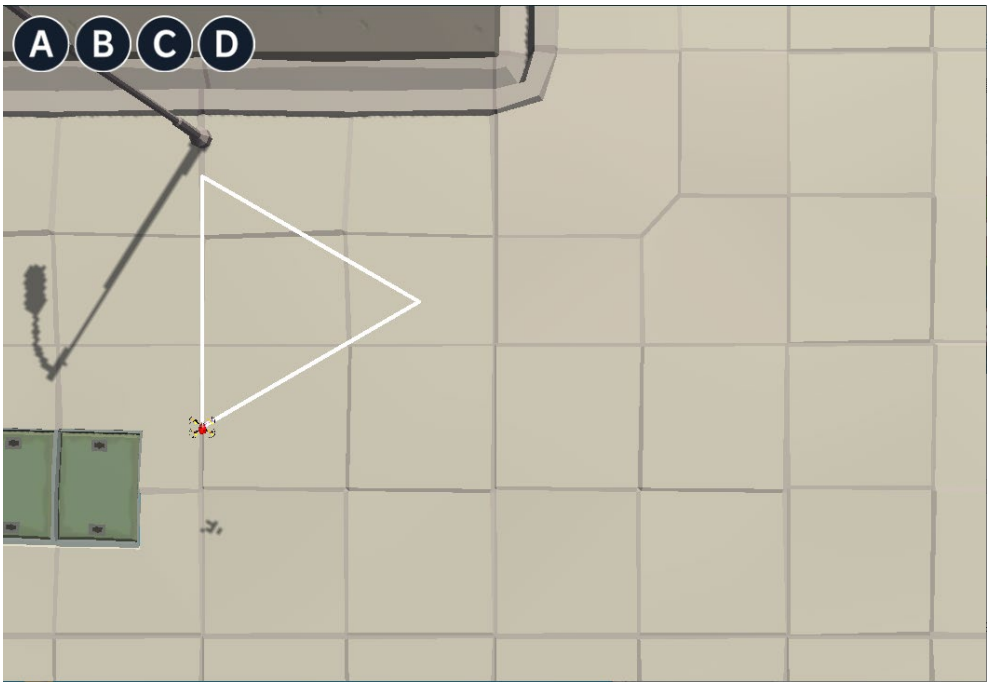
# 7 Real culprit?

Where is the Phantom Thief that the man told the detective about? The rabbit seems to know something. To get the information, make a program to draw the shapes that the rabbit requests.

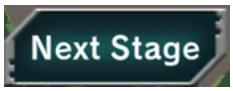

Make the program in the “Drone" object.

Move the drone gradually like an animation to draw lines.




When you have successfully drawn it, tap B to get information from the rabbit about the thief's boss.


We came to a park where the boss seems to be.



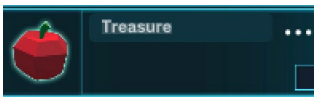

Tap the “Next Stage" button to continue.

## Advanced Challenge

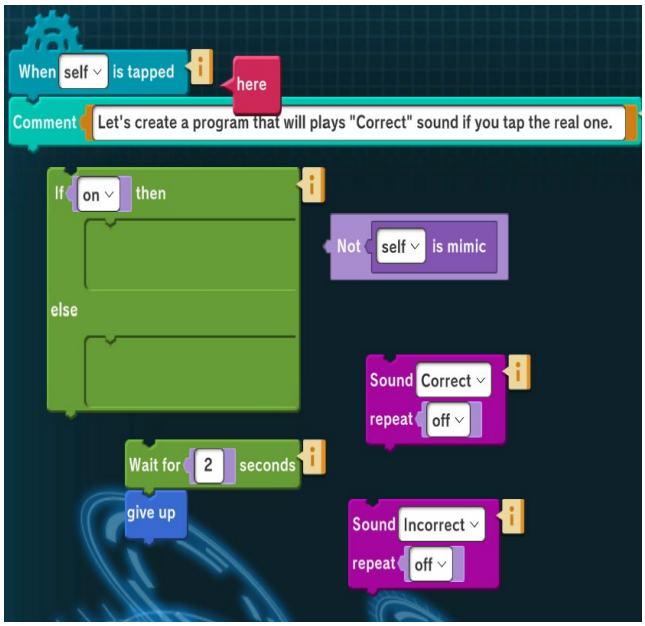
Make the drone draws other shapes such as a hexagons and an octagons. How do you draw a circle?

# 8 The real treasure is

The detective is confronted by the Phantom Thief. The Phantom Thief hides the real treasure among fake ones (alternate). To get the treasure back, you have to find the real one among the fake ones.

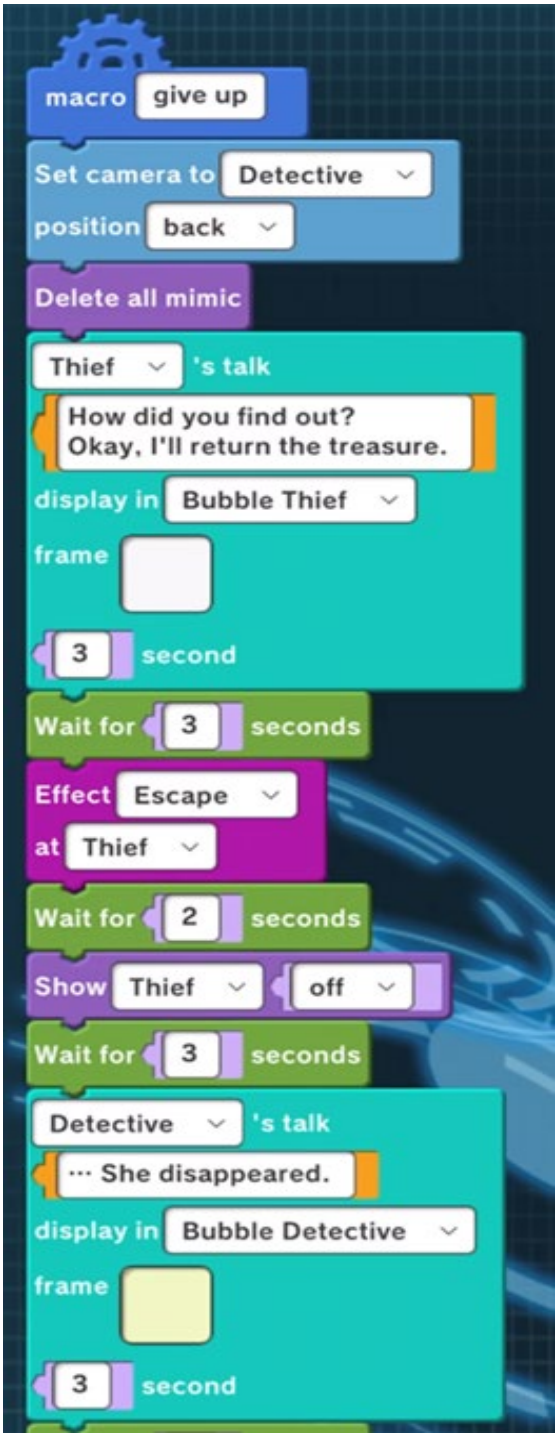




Make a program to find the real one. Make it in "Treasure” object.




Create a program that makes an incorrect answer sound if the treasure is a fake (alternate) and a correct answer sound if it is real.

After you make a program to play sounds, connect "Wait 2 second" and "give up" commands at the end to complete the program.

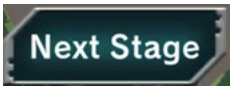



The command "give up" is a new command created using multiple commands. Such instructions are called macros.

What the macro actually does is defined in the program in the object "Phantom Thief".

The figure on the left shows the "Phantom Thief" program.

It defines the Phantom Thief's lines and actions after the real treasure is discovered.




If you are successful, tap the “Next Stage" button to proceed.

## Advanced Challenge

How can you incorporate rules like "the detective can touch the apple maximum three times" into your program? Let's modify the "Treasure" program to do this.

# 9. Secret of the treasure!

After retrieving the treasure, the detective visits his client, Marie, again. Then the detective learns the secret hidden in the treasure


This concludes the story of the detective story. How did you like it? All the programs that make the story are created with command blocks. Feel free to change the program to create a more interesting detective story.

## Advanced Challenge

Once you are comfortable with programming, take a look at programs beside assignments. Modify the program to create an original story. For examples,

- Change the characters
  Note: You may need to adjust the size of the characters.
- Replace effects and sounds to whatever you like.
- Feel free to change or recreate the story as you like.

# 7. Let’s make a game

## Let’s make a game

“Forest Girl” was a fun story and fun content。Using Mind Render you can create a lot of content beside stories. There are games created using Mind Render in “Let’s make a game”.

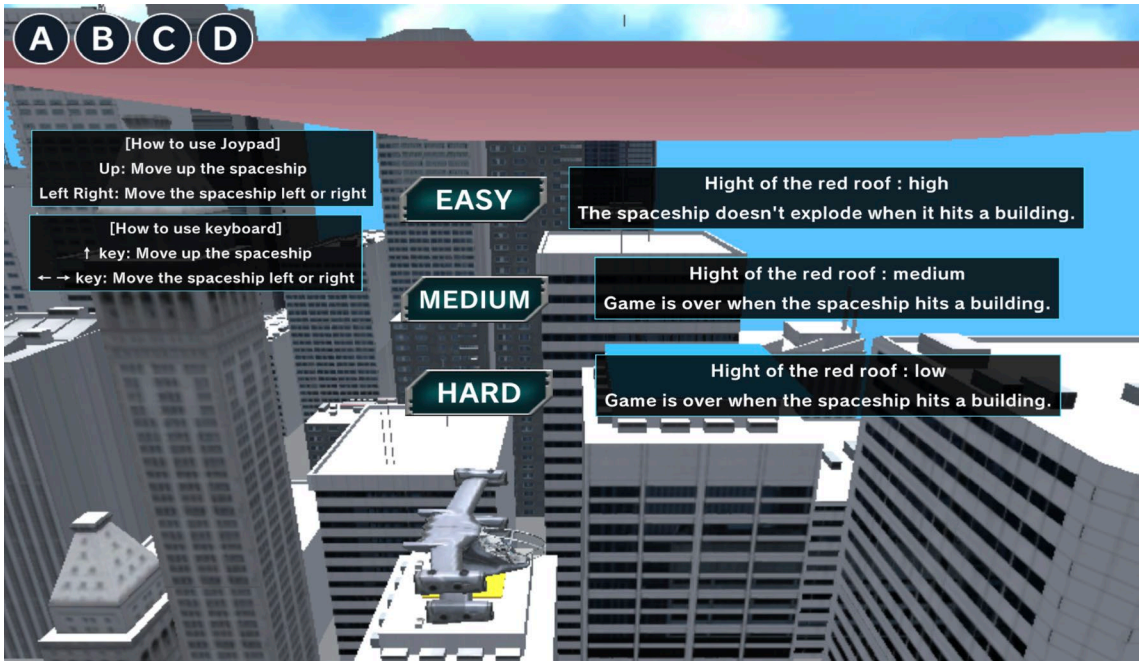
Let's take a look at the first game by choosing “Let’s make a Game” from the Lab menu.

## Spaceship landing

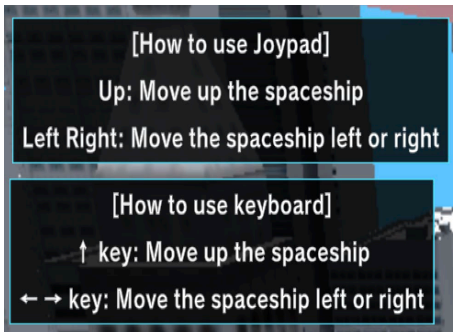
The first game is to control a small spaceship off the top of a building to land it on top of the next building. It's a simple game, but you will find it not so easy until you get used to it.

Tap the "START" button to play the game.


When you run the program, you can select the difficulty level and an explanation of how to operate will be displayed, as shown below. Let's start with the "EASY" level.

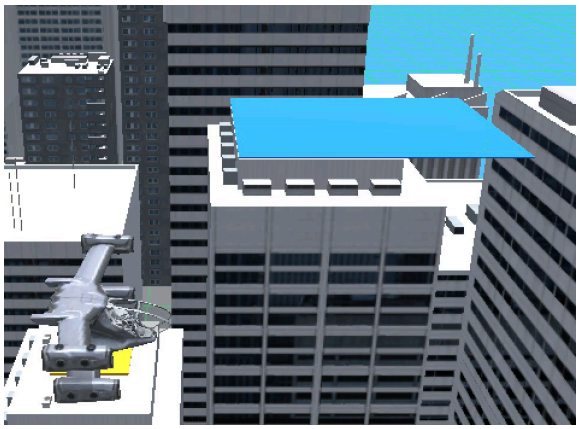



After selecting a level, the game begins. The game is to fly the spaceship on the left to the blue area above the next building by operating the joypad or keyboard and land it. Try until you get used to it.



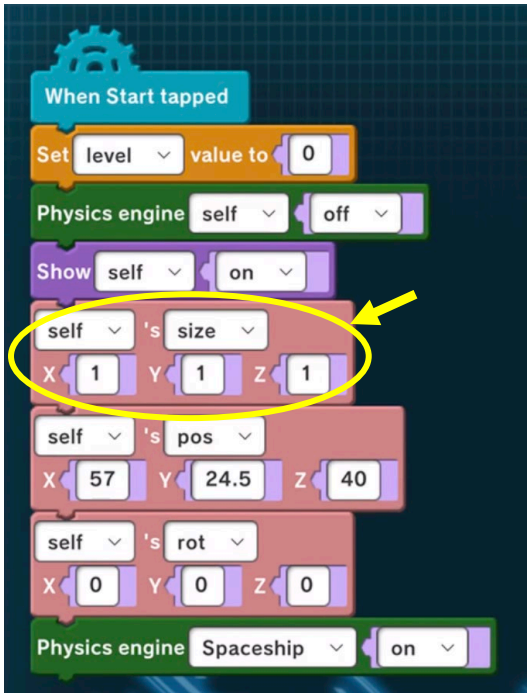

To control the spaceship, use the keyboard if you use a computer, or the joypad if you use a tablet.

Once you complete the (EASY) level, try (MEDIUM) and (HARD) level. In the (HARD) level, there is only a small gap between the starting point and the red roof, so it is not easy to land. Have fun and try until you succeed.




Once you've played with it to your heart's content, it's time to modify the program a bit to make it more interesting. For example, to make the spaceship smaller and easier to control.


First, click Spaceship in the object list. This allows you to view and edit the spaceship object's program.




To change the size of the spaceship, change the number of "your size x (1) y (1) z (1)" indicated by the yellow arrow in the left figure.

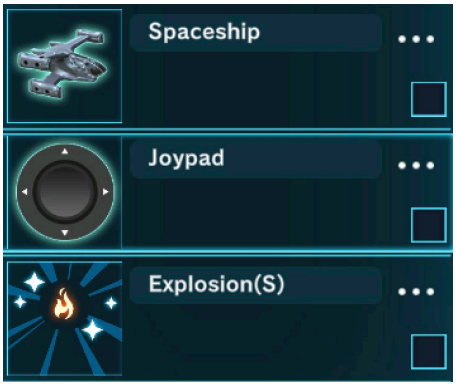
For example, let's see how much smaller the size of x, y, and z will be if each is set to 0.2.


x (1), y (1), z (1)

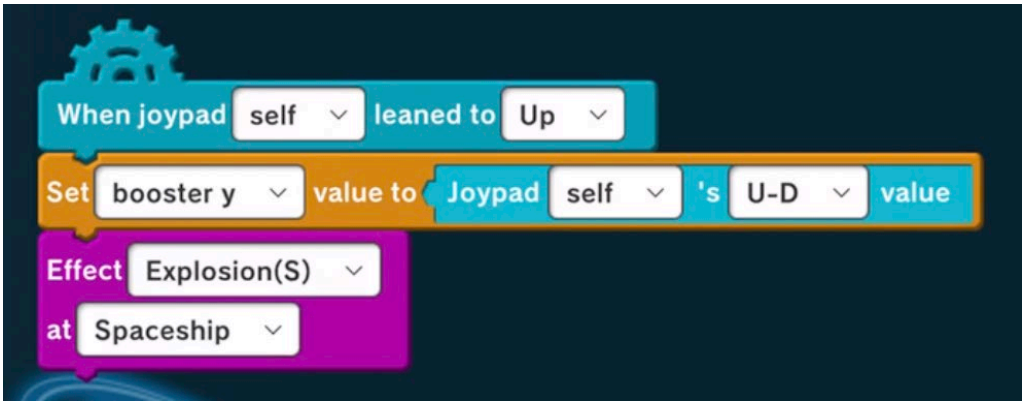

x (0.2), y (0.2), z (0.2)

## Advanced Challenge

How can you change the thrust of the spaceship? The program to control the spaceship is in the Joypad object (for joypad controls).

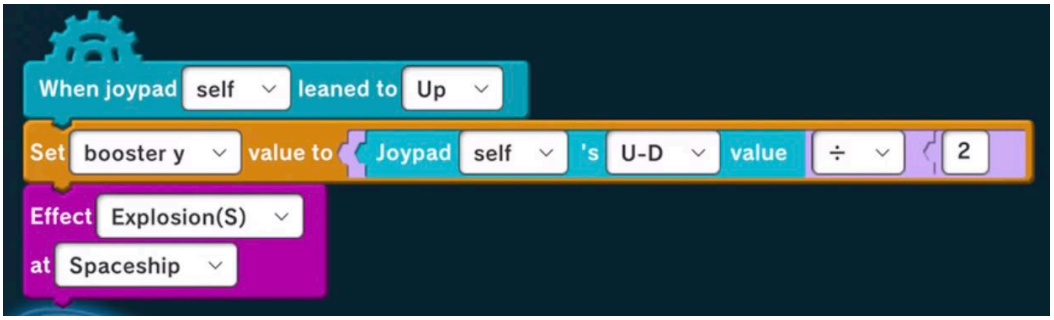



Tap the Joypad in the object list.




For example, to change the thrust to move the spaceship upward, modify the program as shown below.

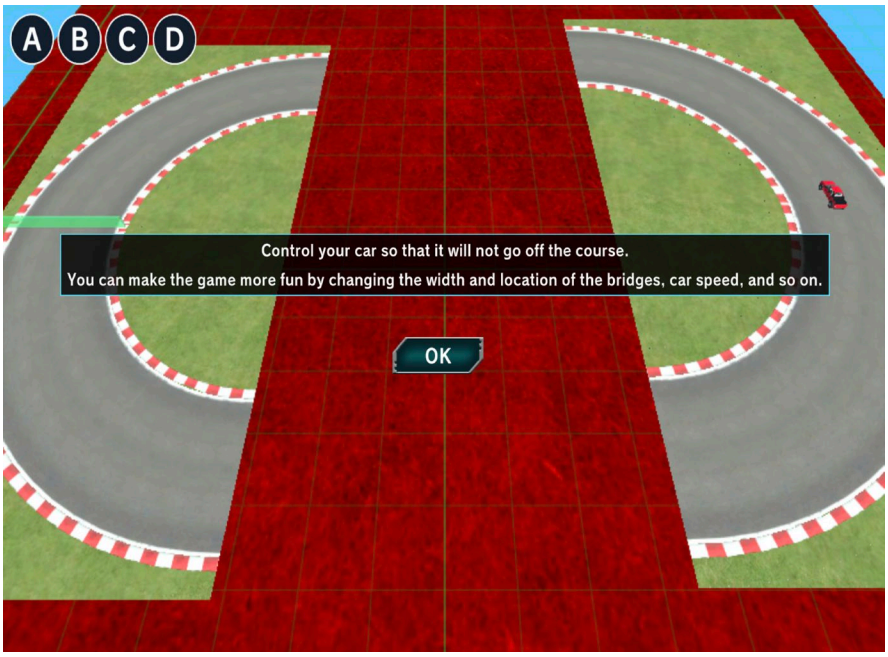
It is the booster command that determines the thrust of the spaceship. The higher the booster value, the larger thrust will be, and the smaller the value, the smaller it will be. In the figure above, the booster value is determined by the up/down value of the joypad. This means how much you move the joypad up and down. To decrease the value to make the thrust smaller, try the following.

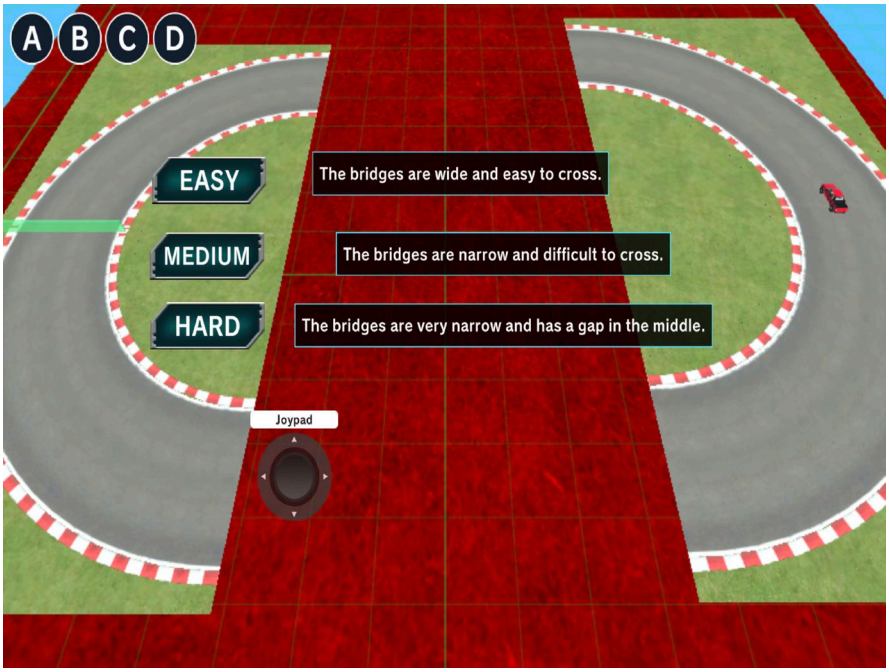



If the number to divide is too large, the thrust will be too small, and the spaceship will not be able to fly upward against gravity. Try different numbers.

# Driving 1

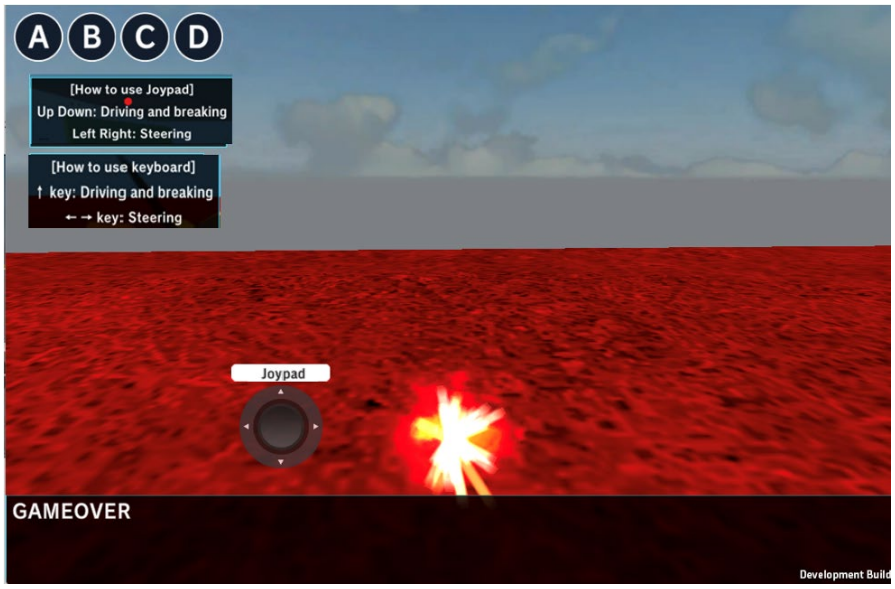
Let’s take a look by tap on “Driving 1”. Initial screen looks like the following figure in full screen mode. Tap the "OK" button to play the game.




There are three levels to choose from: (EASY), (MEDIUM), and (HARD). Use the joypad for controlling.

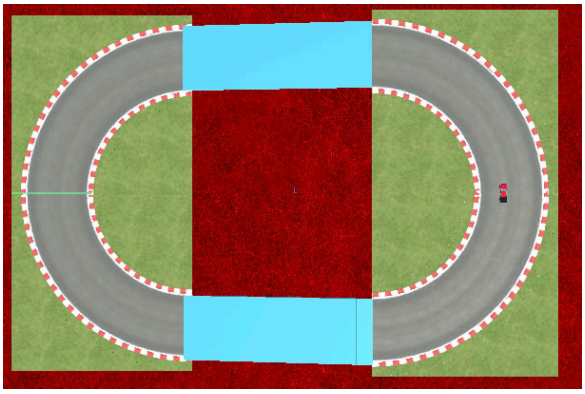
When you select a level and click it, the screen will change as shown below and the game will start. Tap the "START" button and control your car so that it will not go off the course. You can complete the level if you drive around the course without falling off the blue bridge.

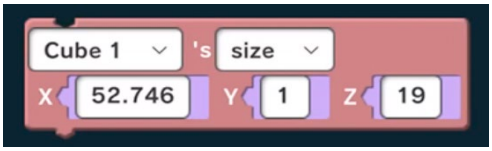

If you fall off the bridge, the car will explode.

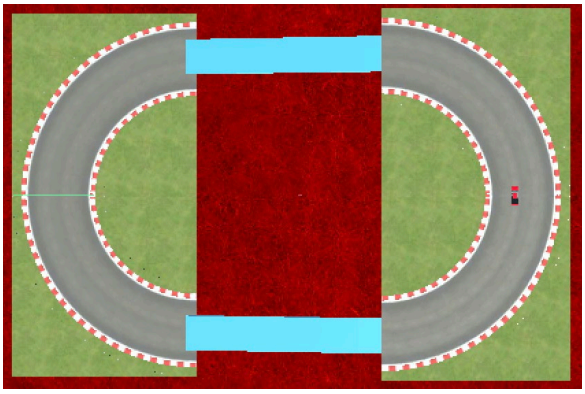
Once you have completed the (EASY) level, you can try to conquer the (MEDIUM) and (HARD) level.

When you're done playing (EASY), (MEDIUM), or (HARD), you can make the game even more interesting. The difficult and interesting part of this game is that you can change size and location of the bridges. As an example, let's see how to change the size of a bridge. The size and location of the bridges can be specified in the command blocks of "EASY", "MEDIUM", and "HARD" in the object list. Below are top views of the course for each level and the corresponding program.

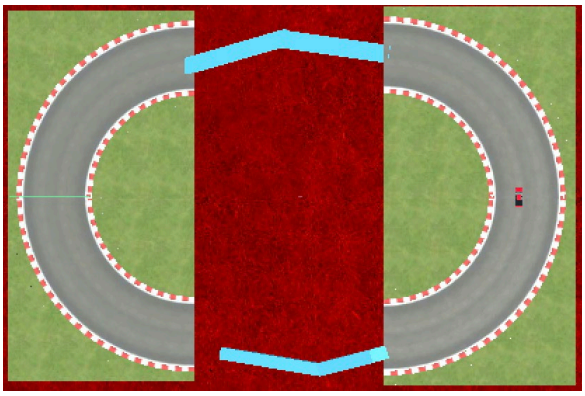


(EASY)

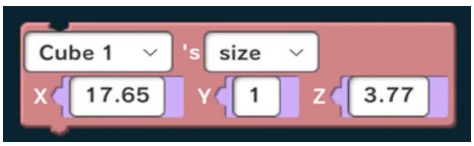


(MEDIUM)

(HARD)




＊＊The two bridges in the (HARD) level are made of two parts each.

## Advanced Challenge

Change the X, Y, and Z values in the program shown above to make the game more fun. Put in various numbers to see what numbers make it easier, and what numbers make it harder, and run the game to see how it is like.

# 8. Let’s study a little 1: “ Let's move the drone“

## Let's move the drone

Let's take a closer look at what you can do with programming using Lab " Let's move the drone ".

## Town

Let's open the lab "Town" by clicking on it from the Lab menu. The initial screen looks like below figure in full screen mode.


Let's run the program prepared in advance in this lab by taping the start button. In this lab, you will enjoy moving the red drone at the bottom center of the screen.

To control the drone, use the two joypads. Tilt the joypad L (located on the left side) up or down to move the drone forward or backward. If you tilt it left or right, the drone will move left or right. Move the joypad R (located on the right side) up or down to move the drone up or down. Tilt the joypad R right or left to rotate the drone to right or to left direction.



Fly your drone around the town, visit different places.

For an example, find the locations ① and ② below by using the joypads.

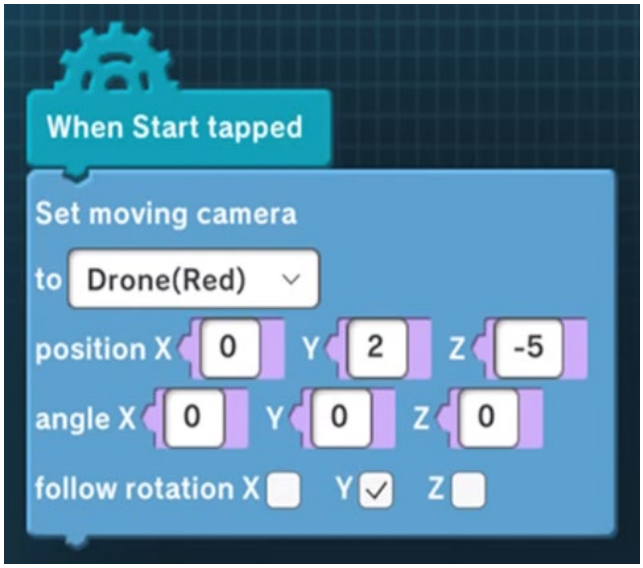
① :

② :


Make a program to switch the camera's viewpoint. Exit the full screen mode and edit the program.

Select the “Camera” object from the object list.


In Mind Render, the camera can be fixed to any place or attached to any object to make it a moving camera. In the initial settings of this lab, the camera settings are as shown in the program below.

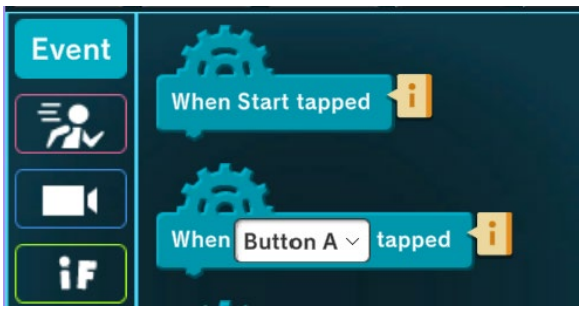



In this setting, the moving camera is attached to the drone. This means when the drone moves, the camera will move along with it.

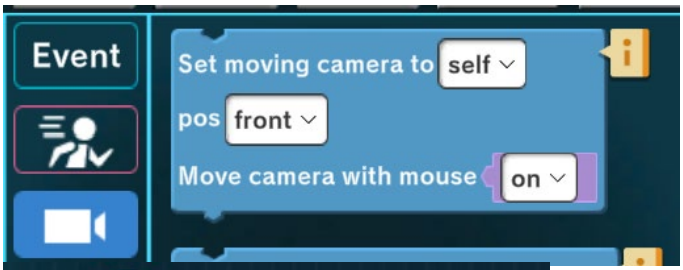
The X, Y, and Z coordinates specify how far away the drone should be attached along X Y Z directions. In this setup the X direction is the position of the drone, Y is slightly higher than the drone, and Z is the location behind the drone.

The left figure shows the view from the camera position when the above settings are made. It is an angle that looks down on the drone from a slightly higher place behind the drone.。

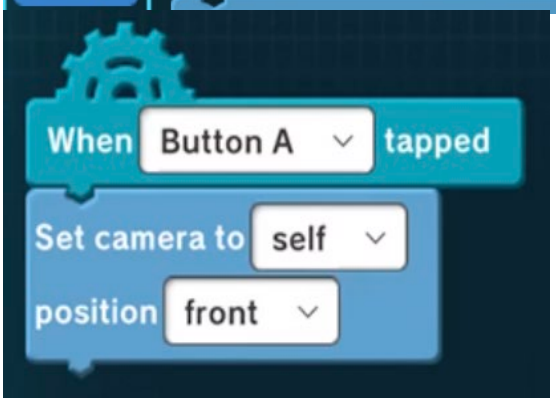

Here, let's create a program to switch the camera using the four buttons A to D.
Make the program in the “camera” object. Tap "Camera" from the object list.



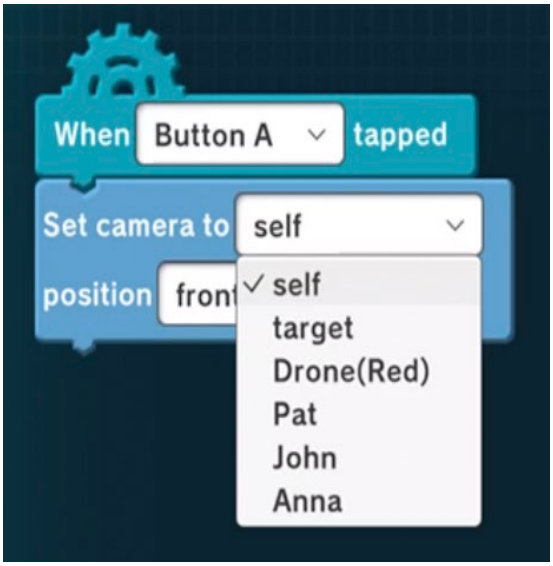

First, click “Events” from the categories. Drag and drop "When Button A tapped" from the blocks displayed in the command block area and bring it to the programming area.



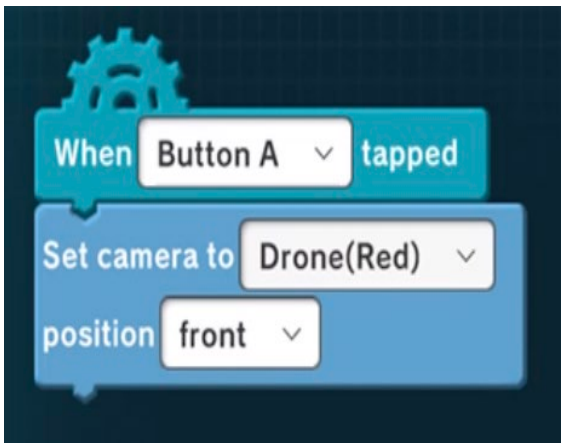

Next, click the camera icon from the category and drag and drop the " Set moving camera to (self) pos (front)" block to the programming area.



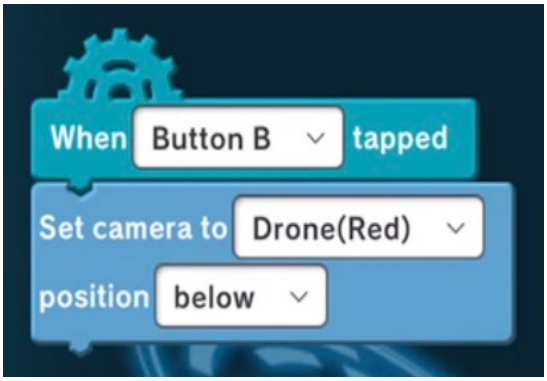

Connect these two blocks as shown in the left figure to create the program.



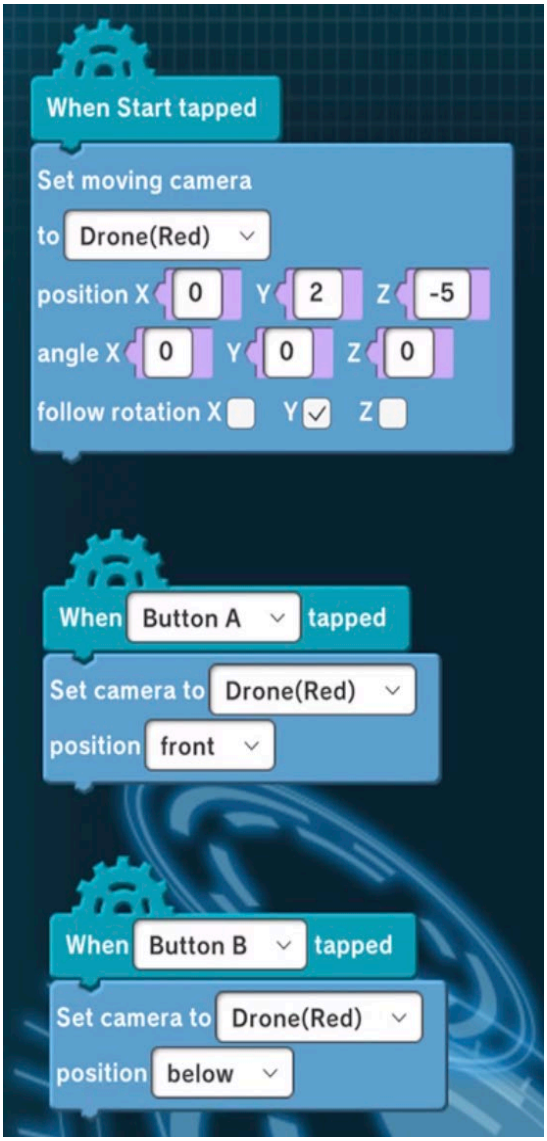

Use the drop-down menu to specify where the camera is mounted.




When the program is completed as shown in the left figure, you can set the camera mounting position in front of the drone by taping button A. The camera will face the front of the drone.

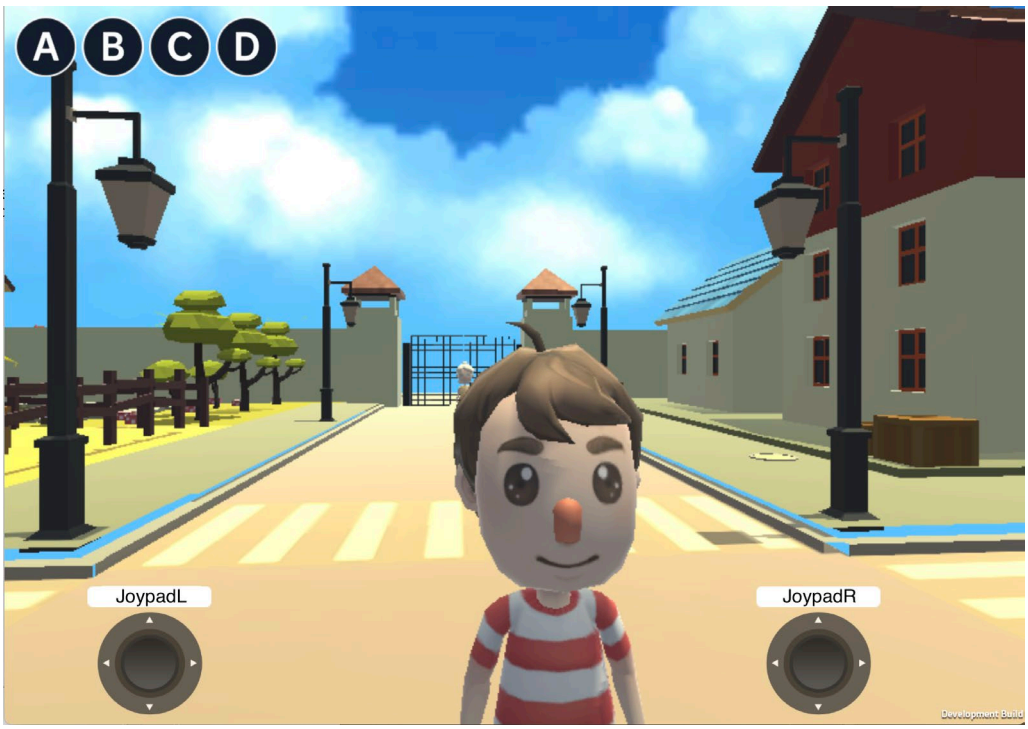



Do the same as above to create the program shown on the left. This program allows you to change the camera's mounting position to the bottom of the drone by tapping button B. The camera will be pointing downward.




By adding the two programs created above, the camera object’s program will look like left figure.

Once you have completed the program, switch to full screen and run it. You can change the orientation of the camera whenever you want.




You can get close to the characters and see a close-up image like this.


You can also see the whole town by going up high in the sky and switching to the downward facing camera with the B button.


Where can you get a view like the left figure? Explore with the drone and find out.

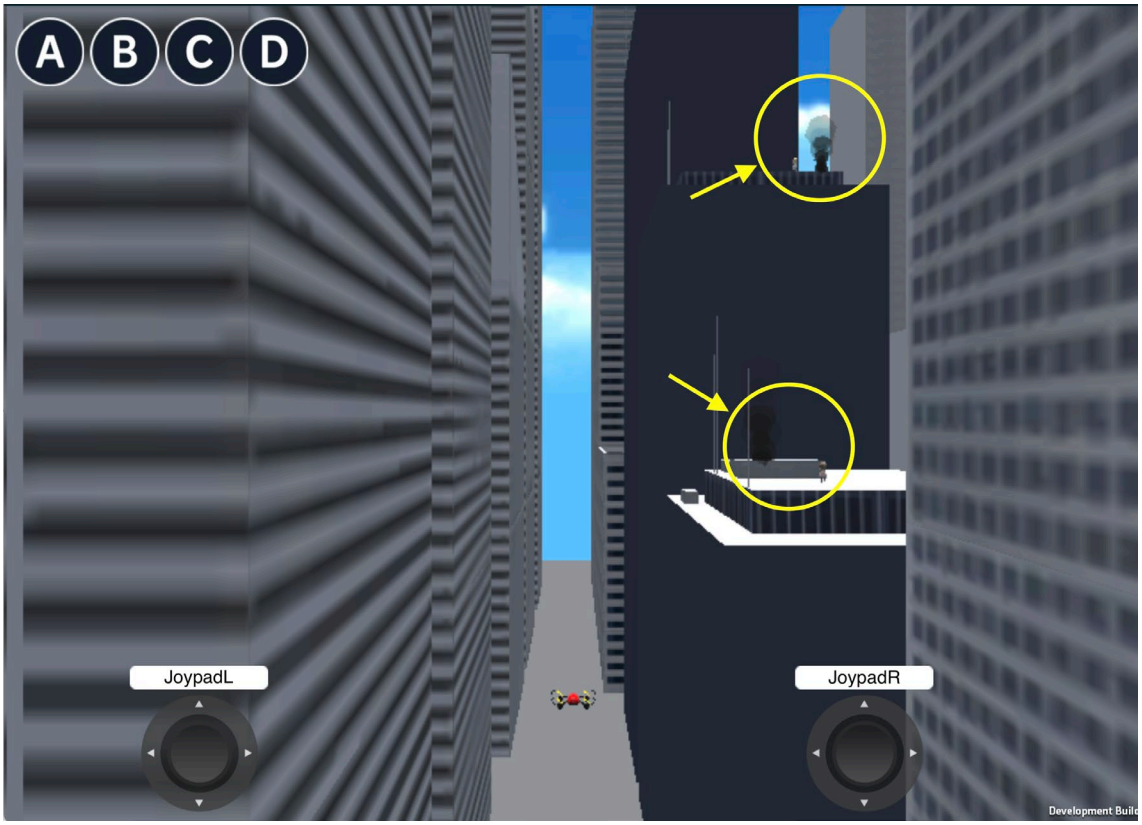
## Advanced Challenge

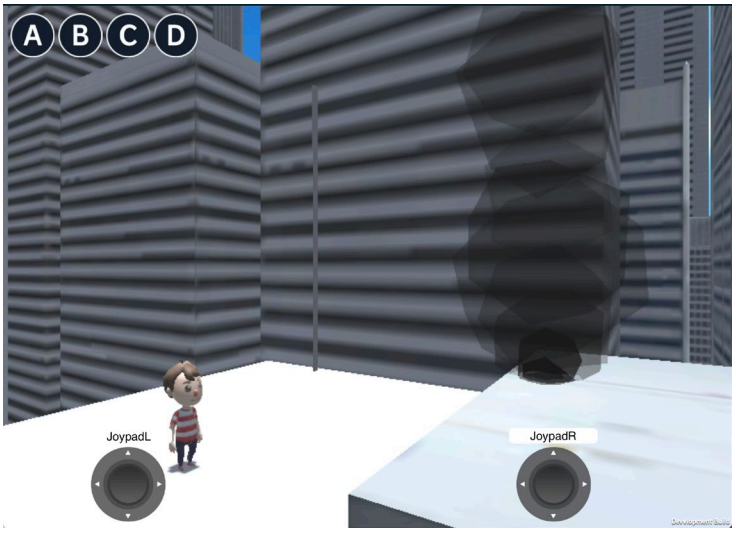
The camera can be attached to any object. Attach the camera to various objects, adjust the angle, and watch the drone fly. For example, how the drone flight would look from one of the character’s view, or how it would look from the street light or from the roof top of a house.

# Life-saving

Lab “Life-saving” challenges you to save lives by flying a drone in a skyscraper city.。

Open Lab “Life-saving” and run it in full screen. You will see the black smoke rising above the buildings as shown on the left.




People are waiting to be rescued in the place where the smoke is rising as shown in the left figure.

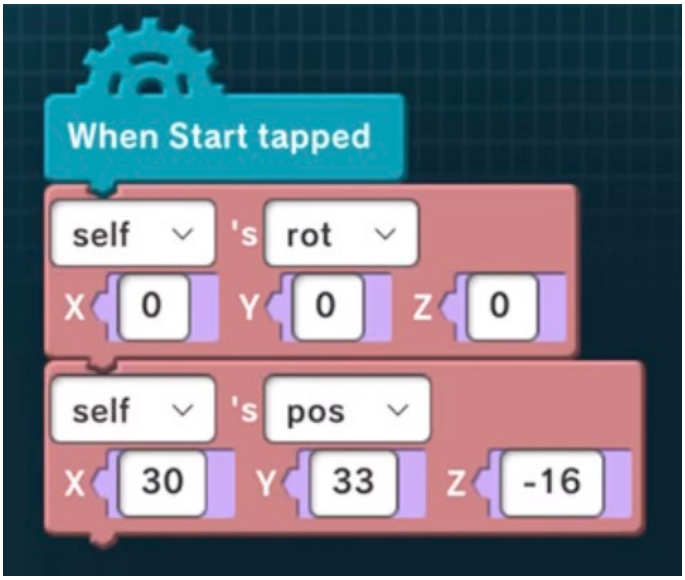
There are four people on top of buildings waiting to be rescued. Find the four places where people are waiting. Use the rising smoke as landmarks.

Let's create a program that automatically visits places where the four people are in order.
The X, Y, Z coordinates of where the four people are as follows.
Pat's location: X (42), Y (31), Z (13)
Tom's location: X (46), Y (52), Z (30)
John’s location: X (40), Y (77), Z (55)
Anna's location: X (38), Y (114), Z (110)

There are many buildings, so you need to move without hitting them. In order to do this, the drone needs to go up to a certain height (Y) so it will not collide with any buildings, and then move to the X and Z positions.

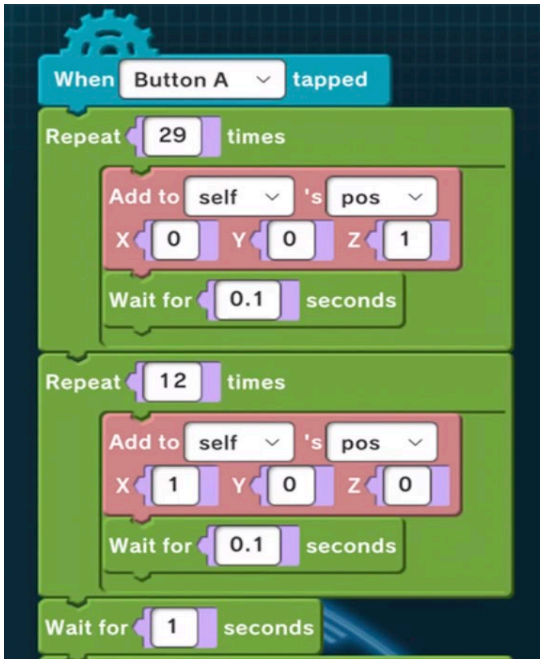

Make the program in the Drone object.

Select the drone from the object list by tapping it.

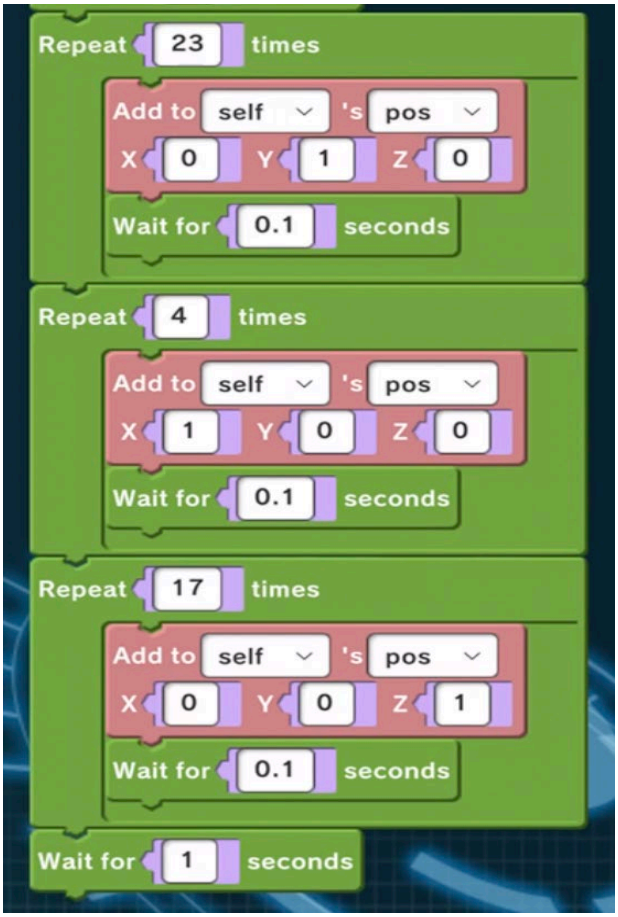



The position of the drone when the program starts running is set by the program on the left.

X (30), Y (33), Z (-16)

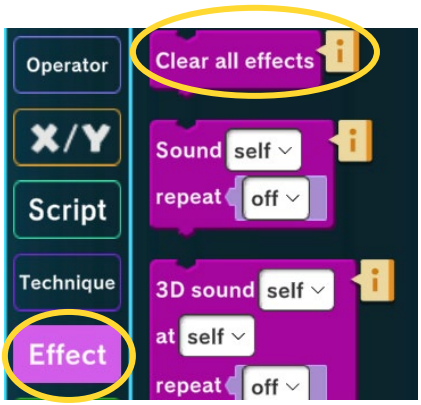



Make "when Button A tapped" a trigger for the automatic patrol program to start. Left figure shows a program where the drone flies from its initial location to Pat’s position. When the drone arrives Pat's position, it waits for one second and then moves on to the next position. Since the drone's current location, Y, is already higher than Pat's position, the drone only moves to X and Z direction. The distance to be traveled is the difference between the current location and the destination X, Y, and Z. Using Z as an example, the drone's current location (at the start) is -16, and Z of Pat's position is 13, so the difference is 29. In the left figure, it moves 29 times by repeating moving 1 at a time 29 times. It is also possible to move the drone momentarily but try to move it repeatedly for short distances so that the movement looks like an animation.




The left figure shows the program to go from Pat's position to Tom's position. Move along Y direction first, then X and Z to avoid colliding with buildings.

In the same way, make other programs to move from Tom to John, and from John to Anna.




When you have visited all four people, let's stop all smokes by adding the command on the left to the end of the program.

## Advanced Challenge

- When you get to a person waiting for rescue, do the followings to show he or she has been rescued.

1 ) Display a message saying "Thank you for rescuing me!

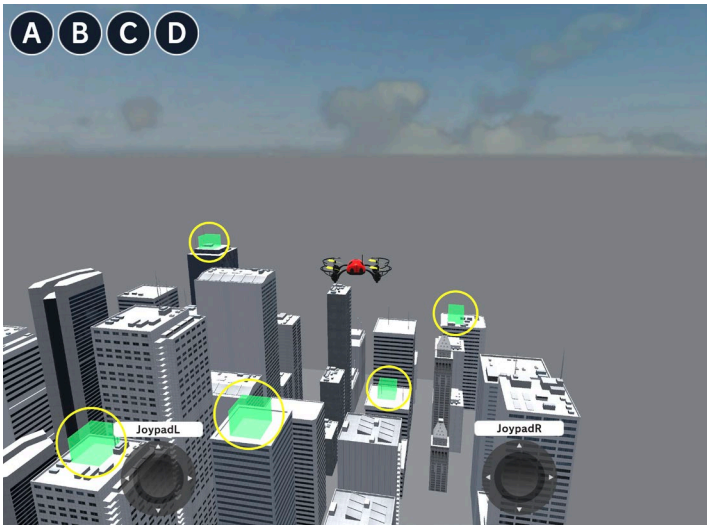
2 ) Once a person is rescued, the person is no longer there. Make them disappear.

There are two ways to display a message: using the "Display" command or using the Bubble object. Try both.

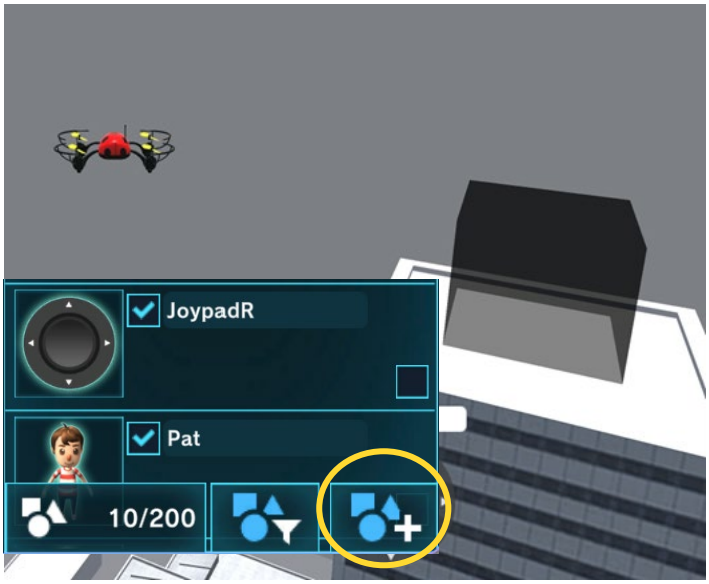
## Drone game

In the Lab “Drone Game”, create a game in which drones hit targets one after another.

When you start the lab "Drone Game", it looks like the figure below. Tap the Start button to start and check around the area.


There are a total of five green cubes on top of buildings. These cubes are the targets in this game.



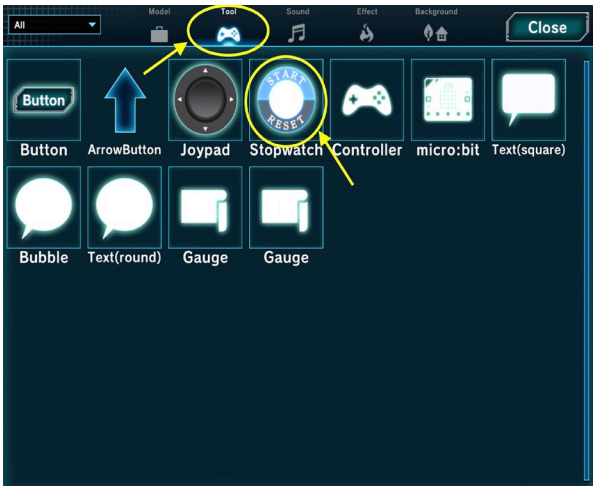

When the drone touches or passes through the green cube, the color of the cube will change to black.

The game is to change the color of all the targets to black in shortest time possible. Have a stopwatch app on your phone and challenge how fast you can complete the game.

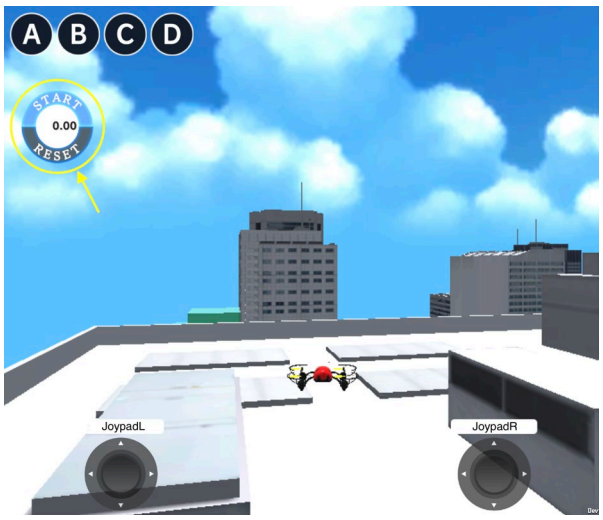
**How to use Mind Render's features to measure time**

Mind Render provides useful tools for measuring time and distance. Use a stopwatch to measure time. Here are the steps to add a stopwatch to the screen.

1）Tap the add object button (the one with the + sign) below the object list.



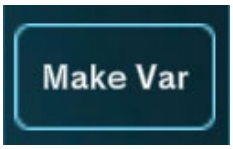

2）Tap the "Tool" tab to display the tool objects, and then tap "Stopwatch" to add it to screen.



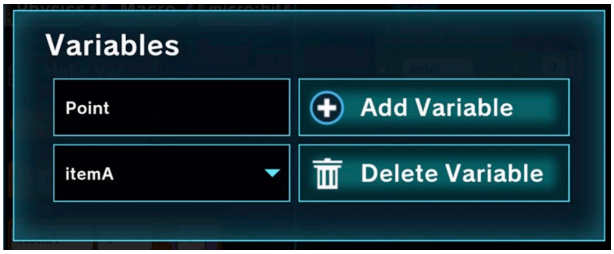

3）A stopwatch is added to the Lab 3-3 screen. Just click Start or Stop and you're ready to go.

Make a program so that every time the drone passes through the target, you get a point.

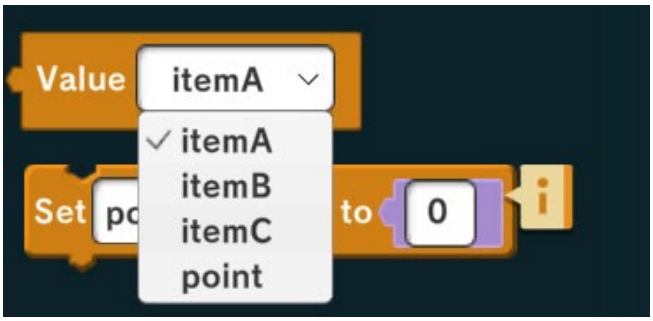
In the object list, there are five cubes. Create a program that adds points to each cube. Let's look at the program for "Cube1" as an example.
First, we need to prepare a variable for the points. A variable is a convenient mechanism that can be used to replace the contents one after another, or to add a number to or subtract a number from the content.



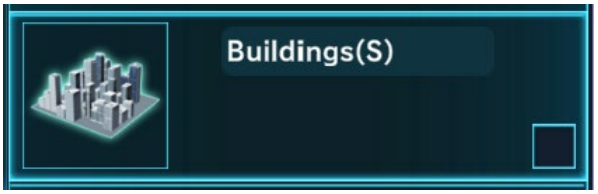

To create a new variable, click "Make Var" at the bottom of the Commands category.



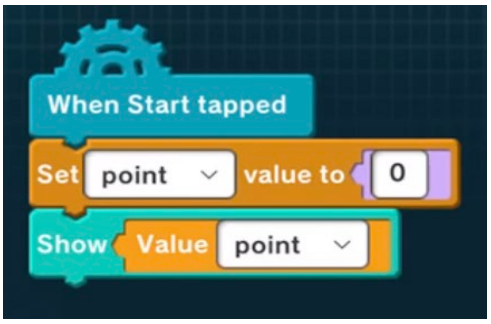

Here we will create a variable with the name “point".




Find the orange global variable command block in the x/y category. Tap the drop-down menu, you will see "point" has been added as a new variable.

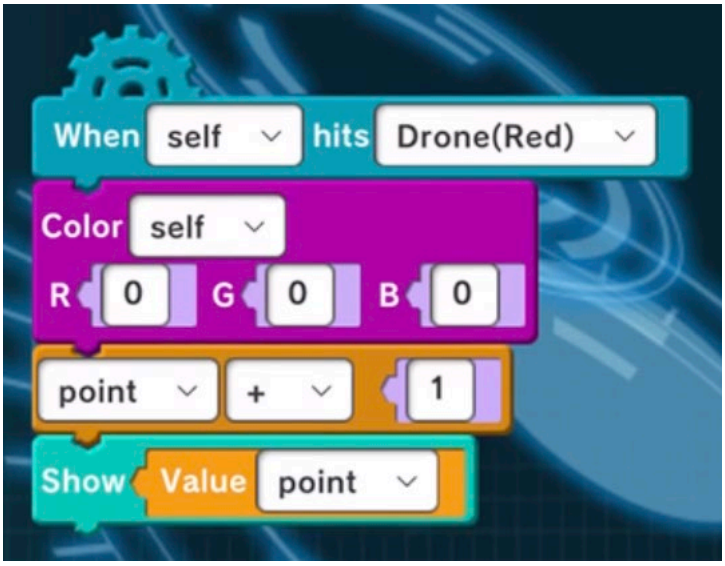



The point variable must be reset to zero when the program is started. This is called initialization. Let's initialize the variable in the Buildings(S) object.



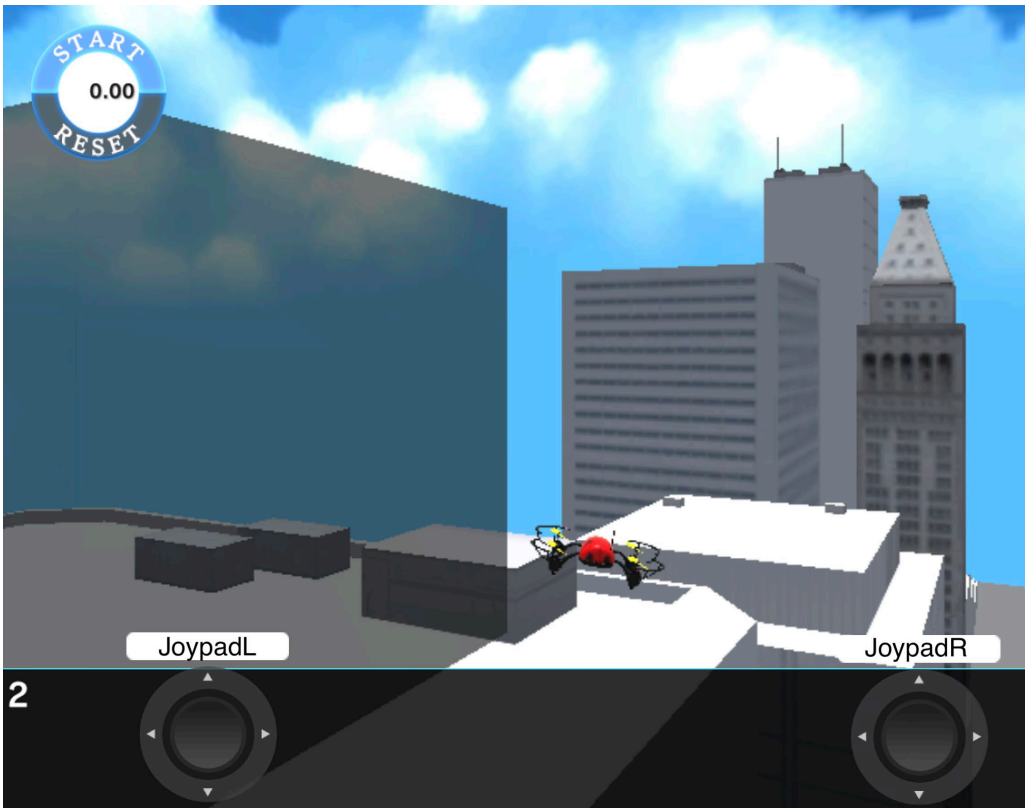

Create a program like the one in the left figure to initialize the point variable. To display the point variable, add the "Show ~" command from the "Script" category and put the "Value (point)" inside.


When the program is started, the display area (black area) will appear at the bottom of the screen as shown in the left figure. The point variable’s value will be displayed there.



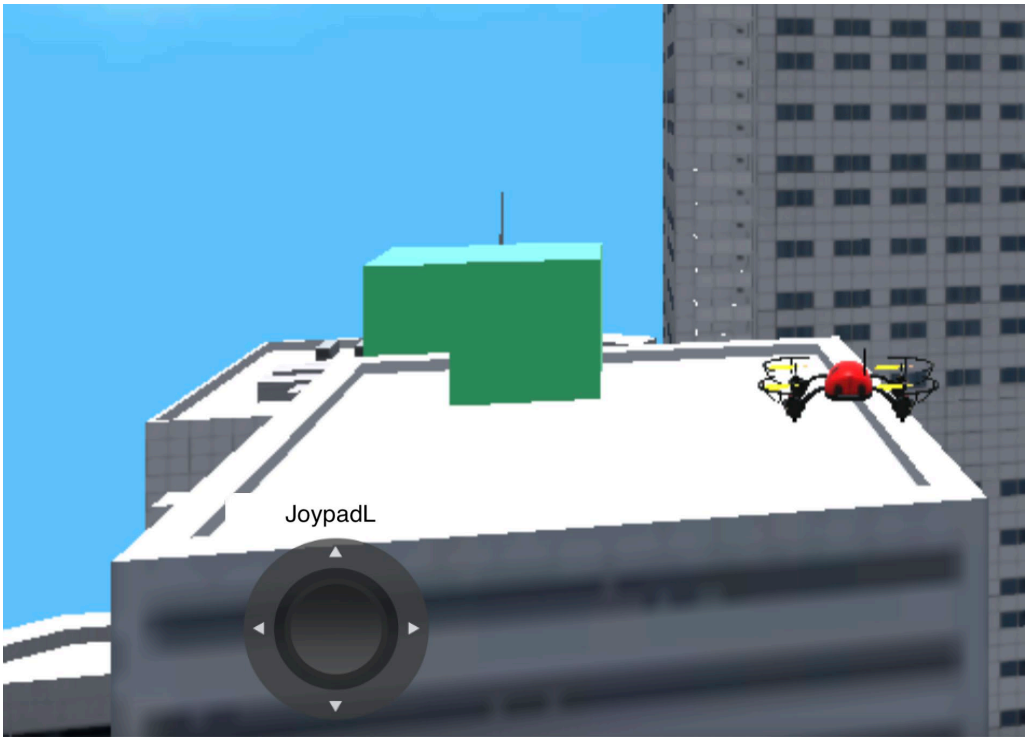

Change the program of the five cubes as shown in the left figure. The purpose of the programs is to increase the points by 1 each time the drone passes the target. It also shows the current point in the display area.

Time and points are displayed and it has become more like a game. Instead of adding 1 point to any target you add, try to make the game more fun by increasing the points for targets that are far away or difficult to access.



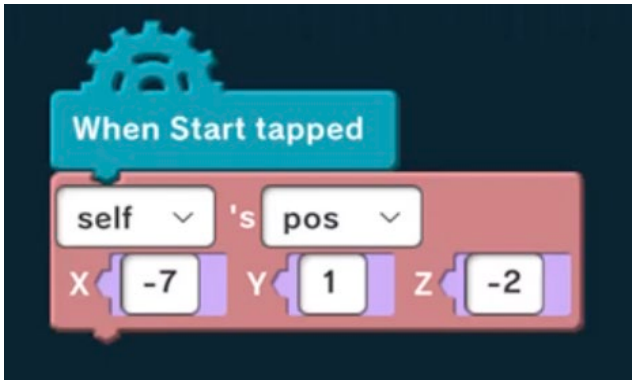
### Advanced Challenge

To make the game even more interesting, try making the target grow in size, shrink in size, or move around a certain distance.




## Make a mechanism

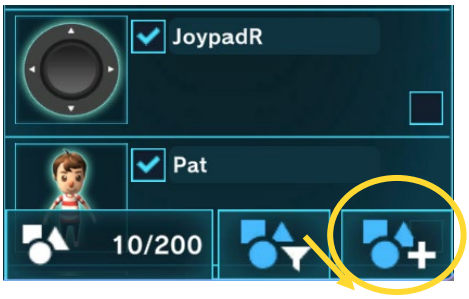
In Lab “Make a mechanism”, make some mechanisms in the background where the drone flies around.




First, set the position of the drone when the START button is tapped.

The left figure is an example of the command settings. Even if not in this way, try various positions and decide by yourself. The figure below shows the view from this position with the moving camera attached to the drone.

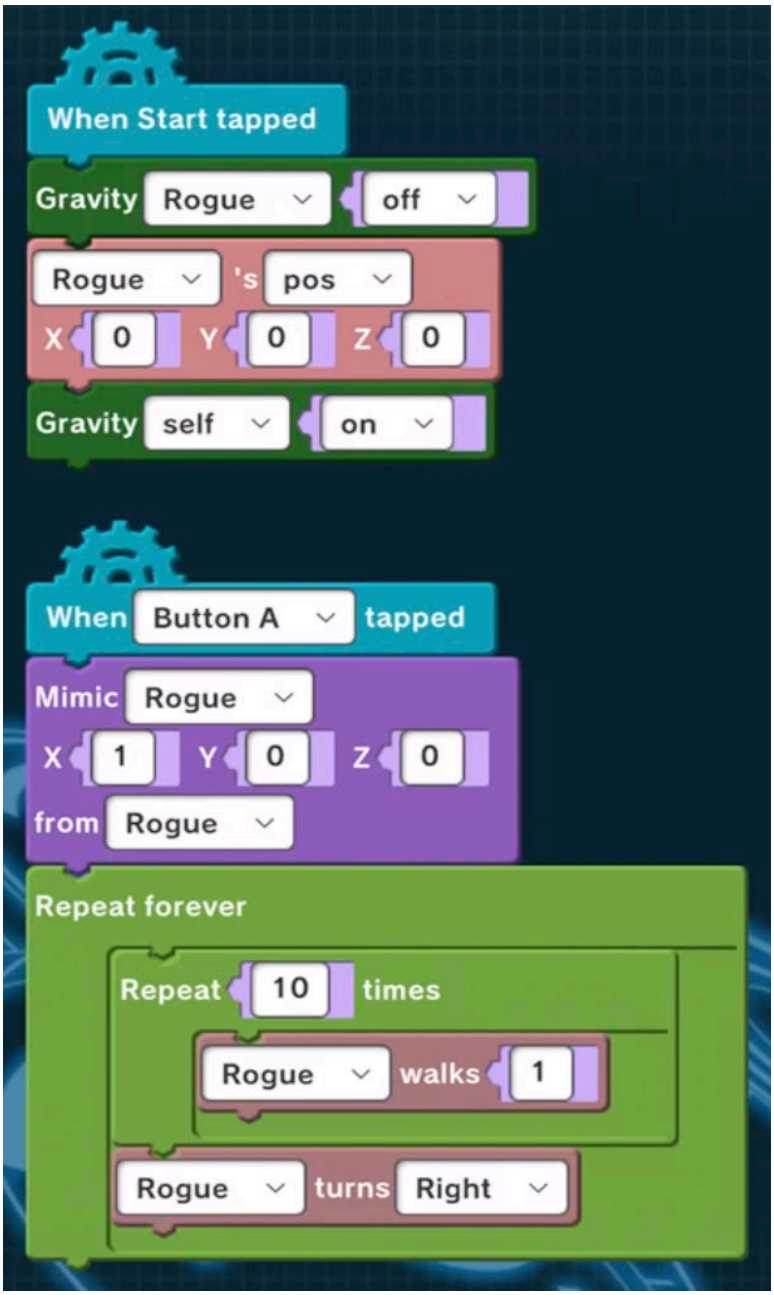

The joypads are the same as Lab “Town.” Make by yourself referring to “Town.” The character in the middle is the one added in the object list.




To add an object, tap the icon pointed by the yellow arrow in the left figure.


Various types of objects such as models, tools, sounds, etc. can be found. Click on the object you want to use, and it will be added.


The added object will be displayed in the object list.

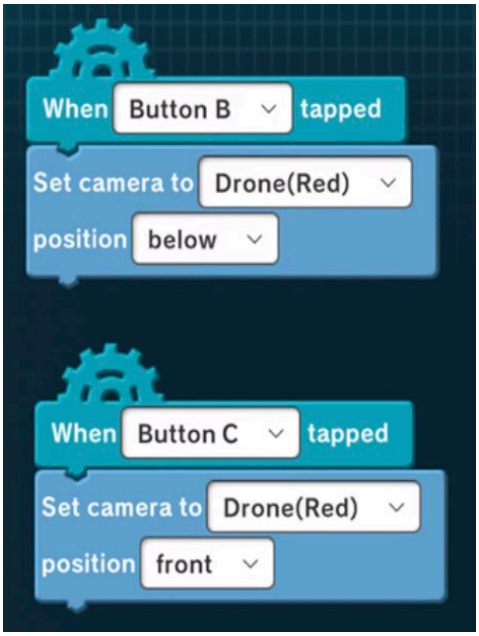



Let's see how many characters move around from various places with a drone. You can add any number of objects using the method shown above, but let's try a more comfortable and quick way to increase the number of characters by programming.

If you create the program on the left at the character you added, every time you tap button A, a copy of the character will be created.


If you tap the A button repeatedly, you can create many copies of the character as shown in the left. The copies will start walking like the original.

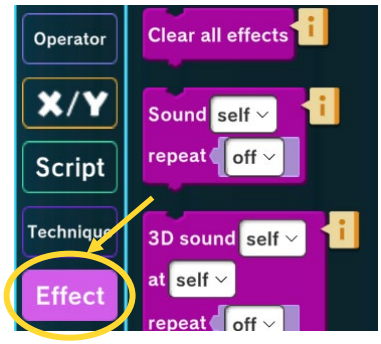
Let's try different views by making it possible to change the angle of the camera by programming.

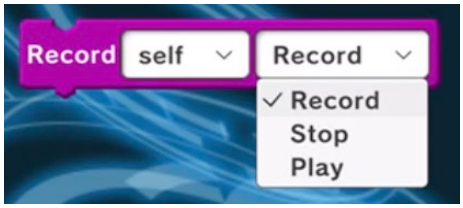

For example, if you add a program to the camera object as shown on the left, you can switch the camera orientation at any time by simply tapping B or C button.

## Challenge movie production

Let's try various things to see if we can recreate a movie-like scene while flying the drone. Mind Render also has commands for recording and recreating the movement of objects controlled by the joypad.




You can play back the movement of the object by using "Record (Record/Stop/Play)" in the "Effect" category.

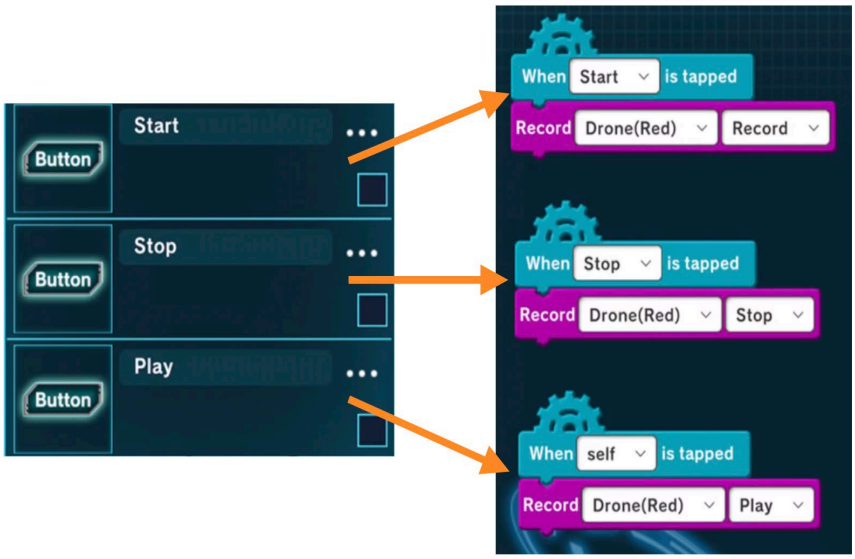



I would like to use a button for the operation that triggers the use of this command, but the A, B, and C buttons have already been used up to this point. Don't worry, Mind Render also lets you add buttons. Let's open the tool tab by taping the add object button.


Tap the "Button" object in the upper left corner to add it to the object list. In the same way, add two more buttons. (Three buttons in total). Each button will be used to start, end, and play the recording.


Once the button objects are added to the object list, place the cursor over the area circled in yellow in the left figure. Type in "Start" for its label. In the same way, type in "Stop" and "Play" for the other two buttons.


As shown in the left figure, three buttons will appear on the screen. You can adjust the position of the buttons to make them easier to see.




Create a program for each of the three buttons as shown below.

Use the three buttons to record flying scenes such as aerial shots or flying through the side of a building.


**Note:**

1. The maximum length of time that can be recorded using Start/Stop/Play is 2 minutes.
2. Only the movements of objects that can be controlled with joypad will be recorded. For example, the movements of the character “Rogue” or duplication process of the character will not be recorded.
3. Once a movement is recorded, it will be overwritten if you record it again. You can play back the movement as many times as you like before it is overwritten.
4. You can switch cameras even during playback. You can also enjoy the view by pointing the camera downward when playing back the movement recorded while looking at the front of the drone.

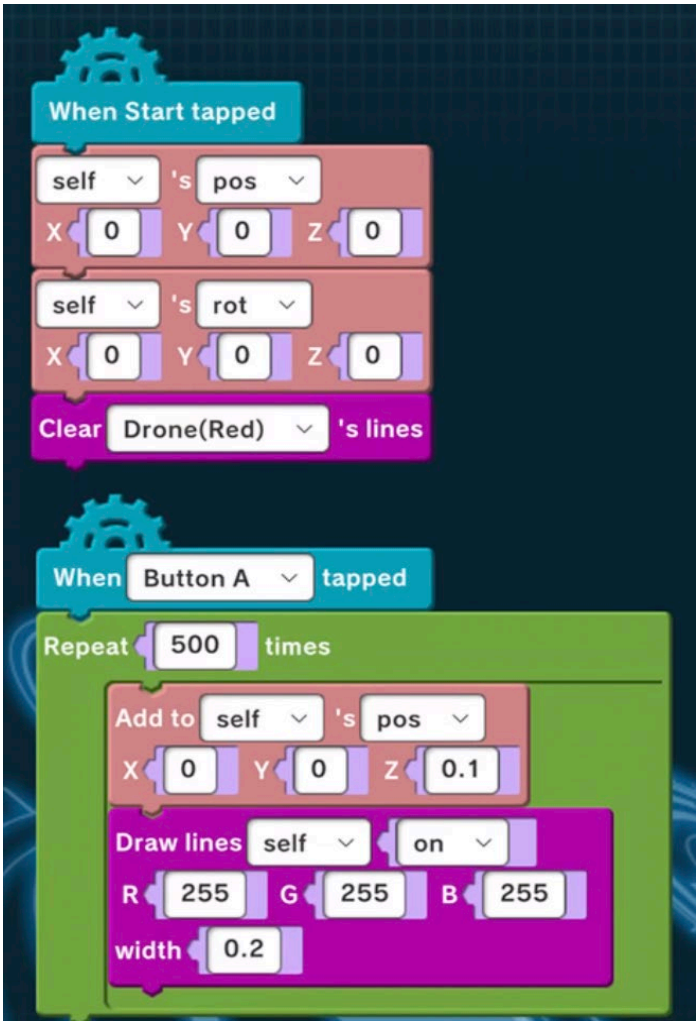
**Advanced Challenge**

Create a short movie by referring to chapter 12. Make an interesting short movie using the backgrounds and objects used in this lab.

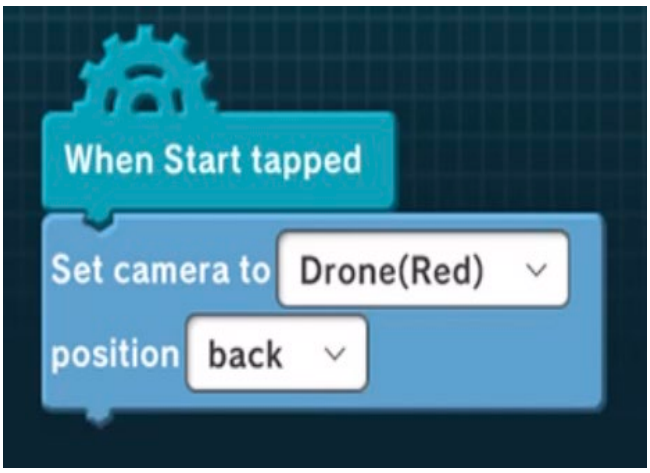
# Draw a path

In the lab "Draw a path", let's make the path of the drone visible by coloring it with a pen. You can draw shapes in the sky, too.


The left figure shows objects to be used in this lab.

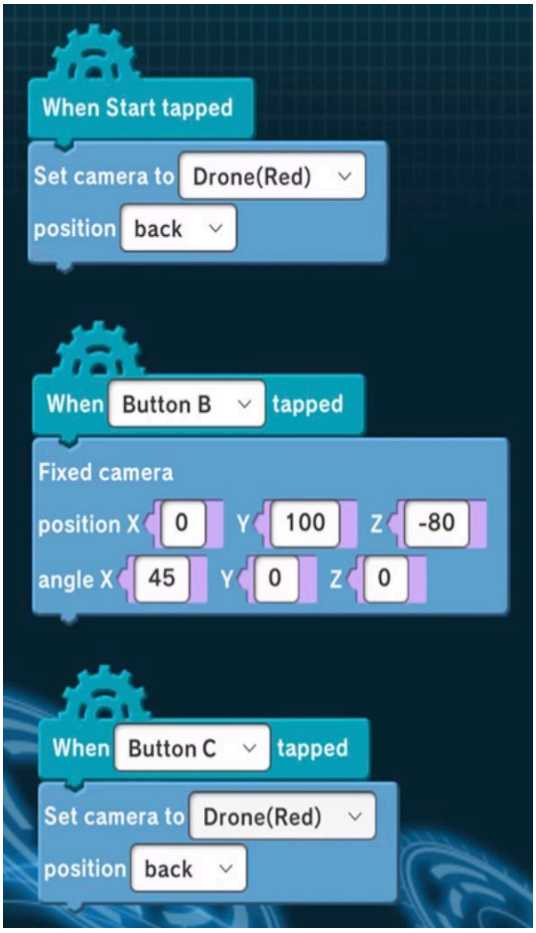



Program for the drone object is shown in the left figure.



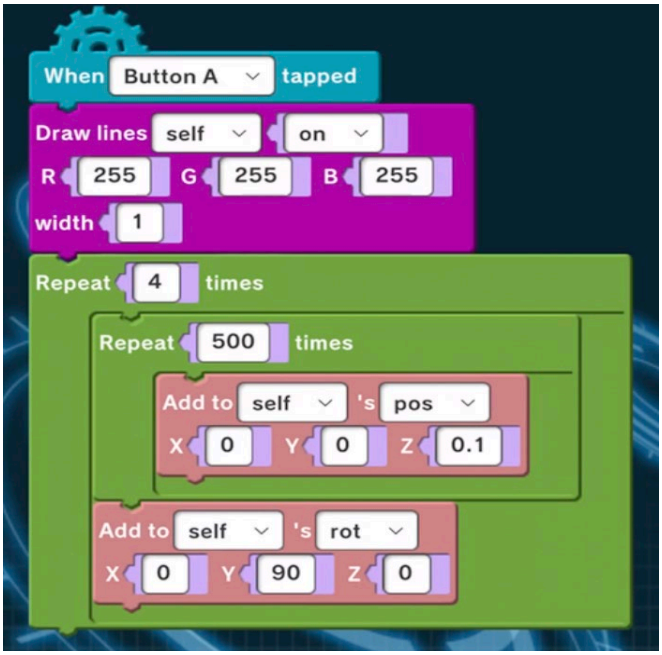

The left figure shows the program for the camera object.

Run the program and tap A button to draw a line as shown in the left figure.




If you put the camera behind the drone, you will not be able to see the line. Make a program that allows you to switch the camera by tapping the buttons as shown in the left figure.

If you look from the sky, you can see the path drawn by the line.



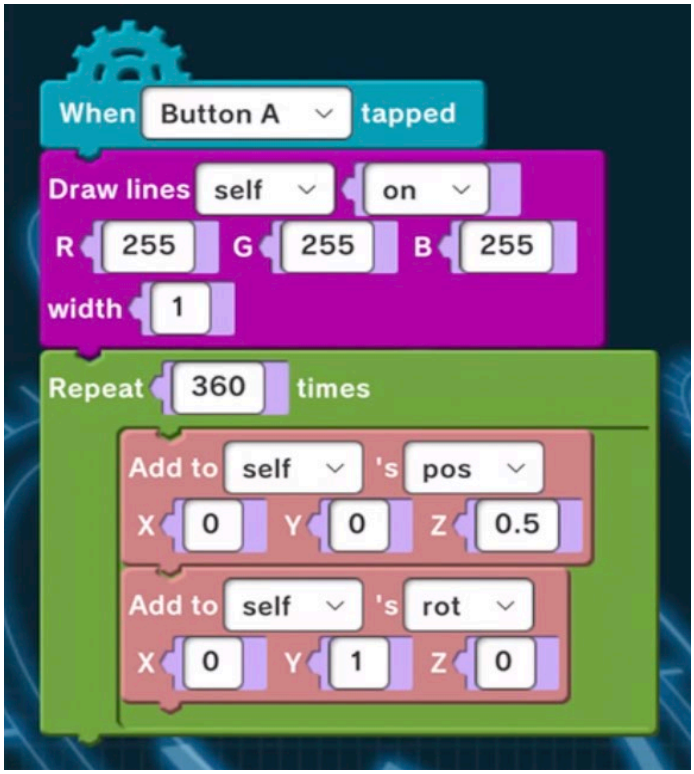

How do you draw a rectangle in the sky?

The figure on the left shows how to do it.

By executing the above program, you can draw a rectangle as shown below.

## Advanced Challenge

How do you draw a circle as shown in below figure?

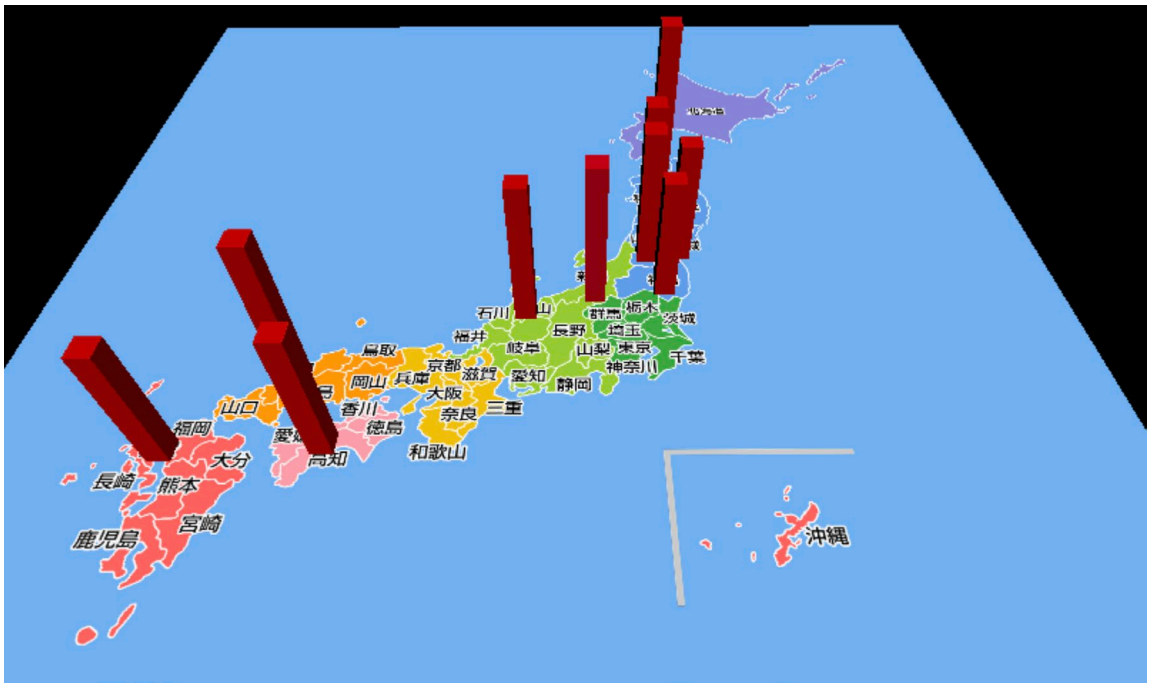



A sample program for drawing a circle is shown on the left. Try it yourself with different settings.

# 9. Let’s study a little 2: “Learn about Japan in 3D“

## Learn about Japan in 3D

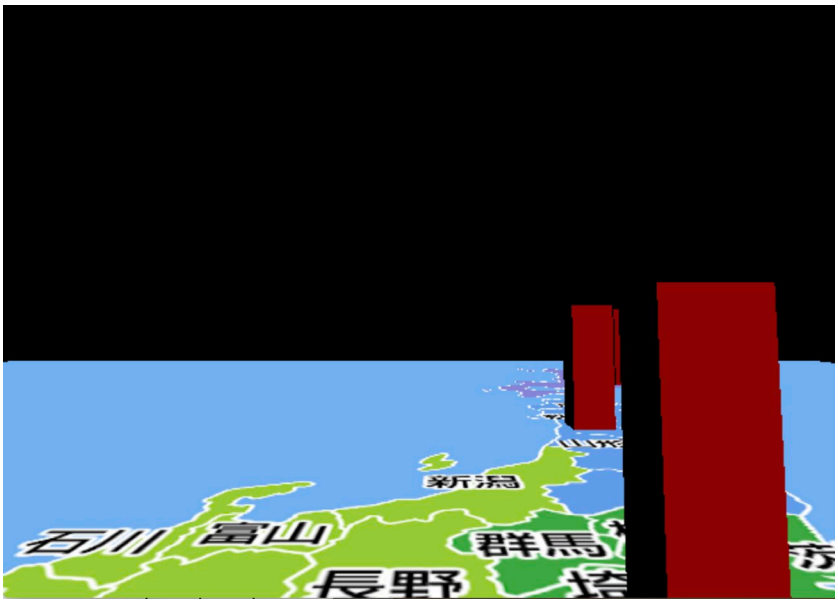
In Lab “Learn about Japan in 3D”, you will learn how to use 3D graphs. A 3D graph is a three-dimensional graph in the form of a cylinder or cube. The below figure shows an example of 3D graphs. Mind Render can also animate how the 3D graphs grow.




Now let's use it. From the lab menu, select "Let's study a little" and then "Learn about Japan in 3D".

## Natto consumption

After opening the lab “Natto consumption”, run the program in full screen mode. After tapping the START button, tap the A button. After waiting for 5 seconds, when the program starts, you can see how the rectangular 3D graph extends over the map of Japan in animation as shown below.




Here, the 3D graph shows the consumption of natto per household in each prefecture of Japan (Hokkaido, Akita, Tokyo, Osaka, and Fukuoka). Let's replace the numbers and see how much money people spend on natto. Statistics used as a reference is the "FY 2008 Report on the Bean Products Promotion Project" published by the Japan Specialty Products Association. This report can be viewed on the Internet. On page 175 of the report, "Annual Per Capita Expenditure on Natto (by Prefecture)" shows how much each person spends per year on natto in each area as follows. The year of the survey was chosen as 2008. The unit is yen/person/year.

Hokkaido: 1,553
Akita: 1,922
Tokyo: 1,458
Osaka: 964
Fukuoka: 1,477
The sample program has been preloaded with the above five data from Hokkaido to Fukuoka.

The figure below shows the command block for drawing a 3D graph over Hokkaido on a map of Japan.

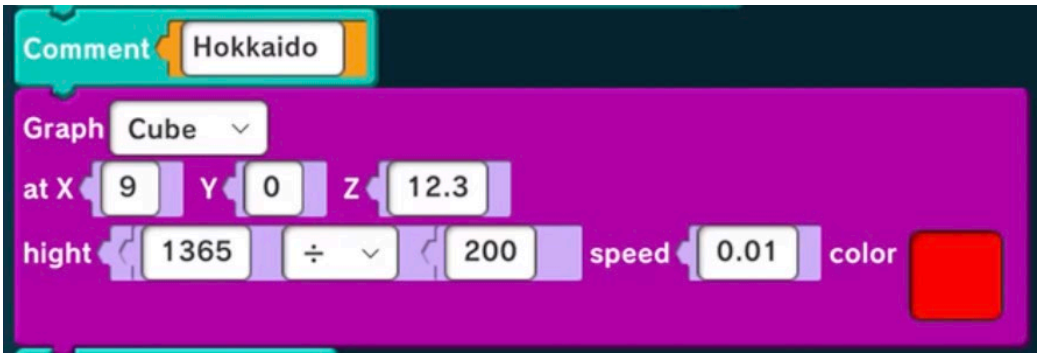
If you tap "Cube", you can select either Cube or Cylinder as 3D graph type in the pull-down menu.

X(9), Y(0), and Z(12.3) specify where on the map the 3D graph should be displayed. Use these three (X, Y, Z) values to specify coordinates to indicate a specific location on the screen of Mind Render. X (9), Y (0), Z (12.3) are the coordinates of Hokkaido on the map.

Height (1553/200) specifies the size (height) of the 3D graph. The maximum value is 10. Adding more numbers will not make the 3D graph bigger. To display a large number such as 1553 in a 3D graph, divide the number to make it fit in the range of 0 to 10.

Speed specifies the speed at which the graph is drawn. If you want to draw slowly, specify a smaller number.

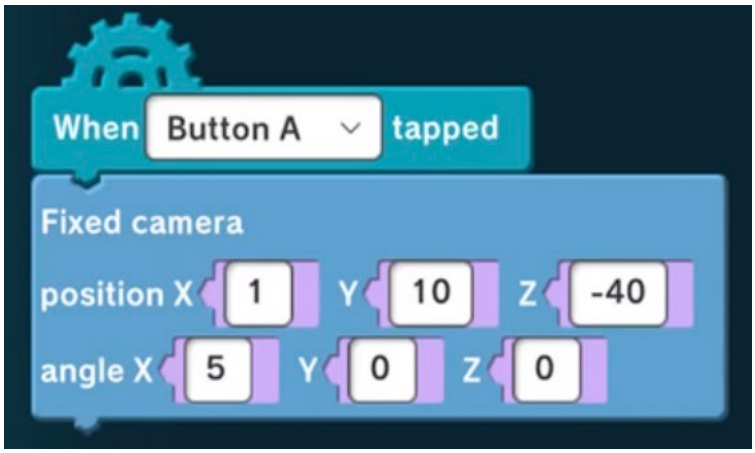
Color specifies the color of the 3D graph. When you click it, a color palette will be displayed, so you can select and specify your favorite color.




Hokkaido has the highest value of 1,553 in natto consumption data. Dividing this by 200 gives 7.765, so you can see that all the data, including Hokkaido, fall within the range of 0 to 10. For Akita, Tokyo, Osaka, and Fukuoka, the original data values are similarly divided by 200. Take a look at the program and see for yourself.

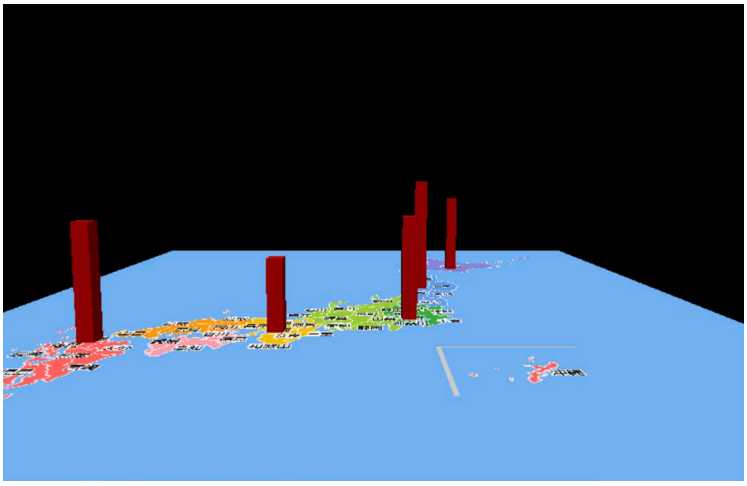
In order to enjoy this program in VR, when the program starts, the camera moves and you can watch the animation as it moves. If you want to view the graph on the entire map of Japan, you can also change the camera’s position following the steps below.

1）Select the camera from the object list




2）Change the content of "When Button A tapped" command as shown below

When you run the program after the changes are made, it is displayed as shown in the figure below.



By changing the value of the fixed camera, the 3D graph can be viewed from various positions.

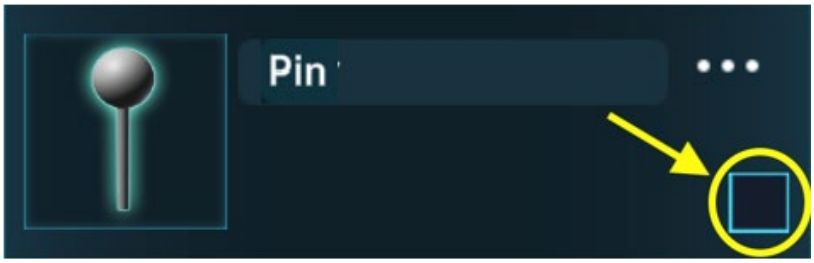
**Advanced Challenge**

On the Internet, you can find a variety of statistics compiled for each prefecture. For example, "Naruhodo Statistics Gakuen" (http://www.stat.go.jp/naruhodo/c1s3.html) introduces various statistics such as population, agriculture and forestry, and energy. Look

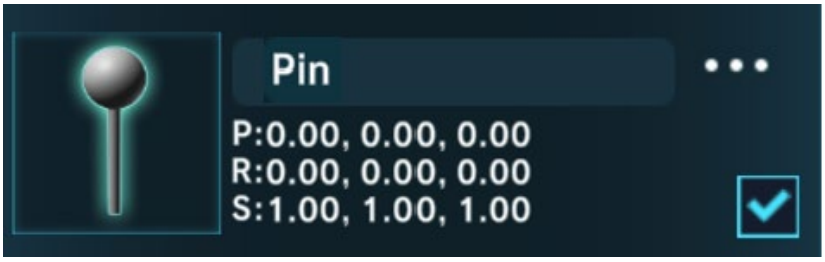
for some interesting statistics in the site of Hokkaido, Akita, Tokyo, Osaka, and Fukuoka, and put them into the program of this lab to display graphs.

## How to draw a graph

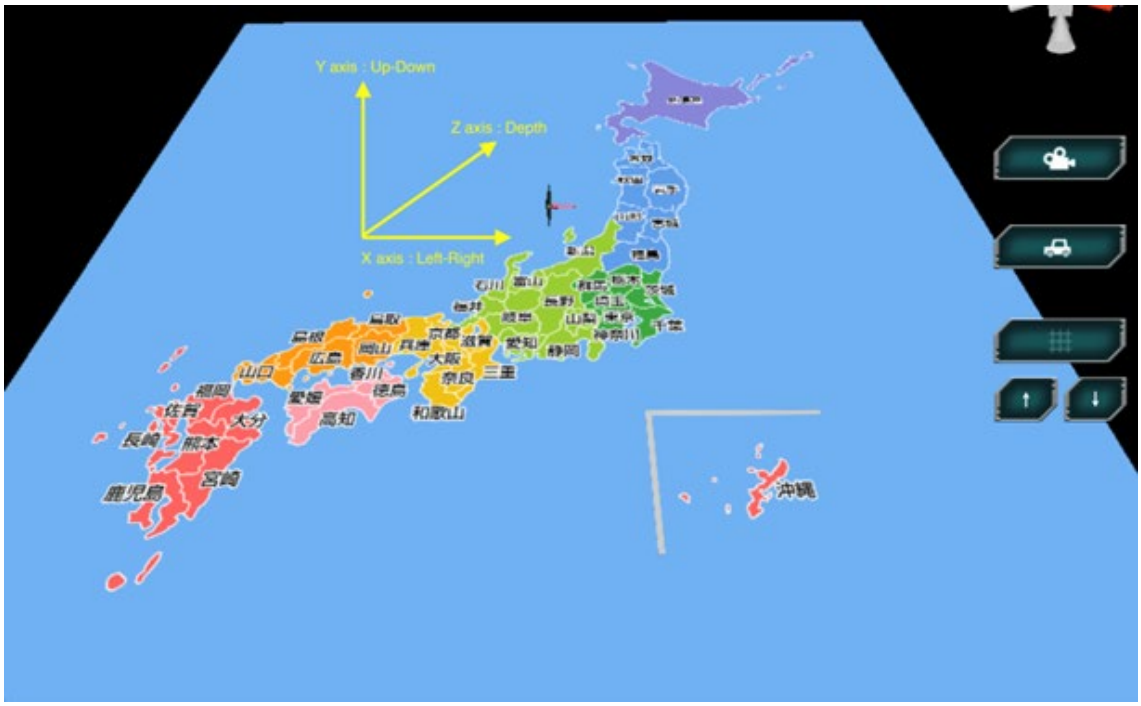
There is no program in the lab "How to draw a graph". In this lab, learn how to identify coordinates in order to draw a graph anywhere you like on a map of Japan.
As a preparation, after opening Lab “How to draw a graph”, add one object. Tap the Add Object button below the object list, select "Material" from the drop-down (subcategory) and tap the "Pin". This pin will be a tool for checking the location on the map. Once the pin is added to the object list, click the yellow circled button in the figure below.




You will see a check mark in the square and small numbers on the left side. The top of this number displayed in 3 lines is the coordinate information.
P stands for Position. Since Mind Render is 3D, coordinates are determined by where it lies on three axes: the X, Y, and Z axes.
P（Coordinate）：X-coordinate（0.00）、Y-coordinate（0.00）、Z-coordinate（0.00）




Please refer to the following diagram to see which direction the X, Y, and Z axes point in the screen.



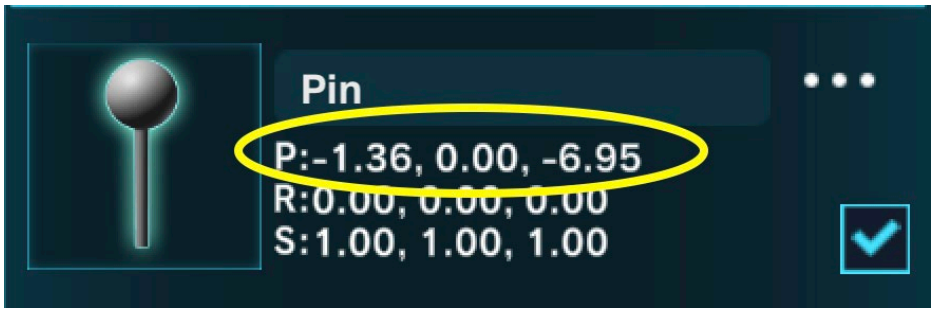

The place displayed when you add the pin is slightly above Niigata Prefecture. The coordinate of the pin is: X (0.00), Y (0.00), and Z (0.00). Please refer to the figure below.

To find out the coordinates of any location on a map of Japan, you can move this pin and check its coordinates information displayed in the object list. The pin can be moved as explained below.

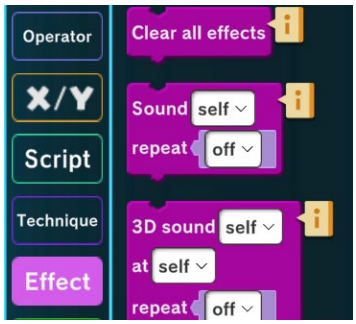

When the program is not running, if you click the pin, three arrows of red, blue, and green will be displayed at the pin as shown below. You can move the pin along the X axis by dragging the red arrow. The blue arrow moves the pin along the Z axis, and the green arrow moves the pin along the Y axis. You can move it well by grabbing the tip of the blue arrow and moving it.

As an example, examine the coordinates of Gifu Prefecture. Move the pin to the location of Gifu Prefecture as shown in the figure below.

When the pin is positioned at Gifu Prefecture, the coordinates of the pin in the object list are as shown in the figure below: X (-1.36), Y (0.00), and Z (-6.95).

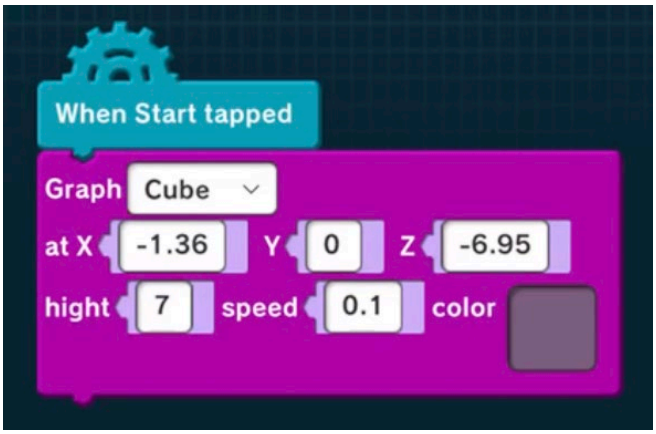



With this coordinate, you can draw a 3D graph on top of Gifu Prefecture. Try to do it.

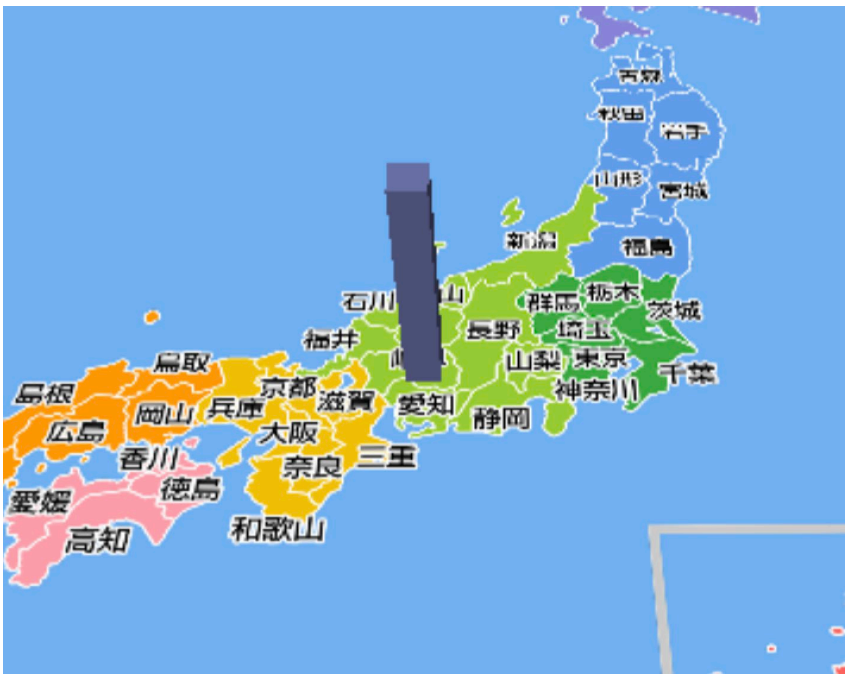



The command block for drawing 3D graphs is in the "Effects" category. The place to create the program is in the Japan map object.

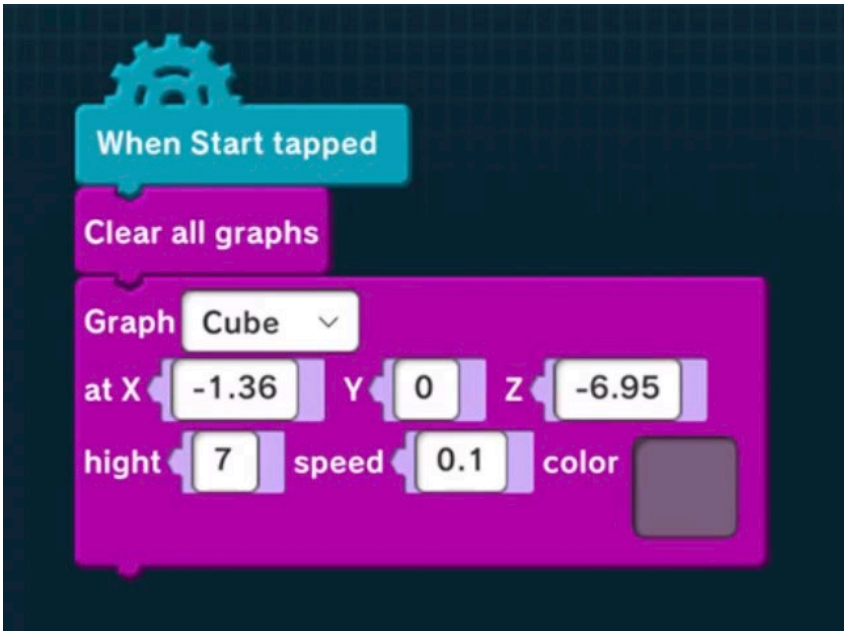
Use the following command block introduced in Lab 4-1. Specify the coordinate information of Gifu Prefecture, size (0-10) and speed (0-10), color, and graph type (Cube: square prism or cylinder). An example is shown below. An appropriate value (7 as an example) is entered for the size.



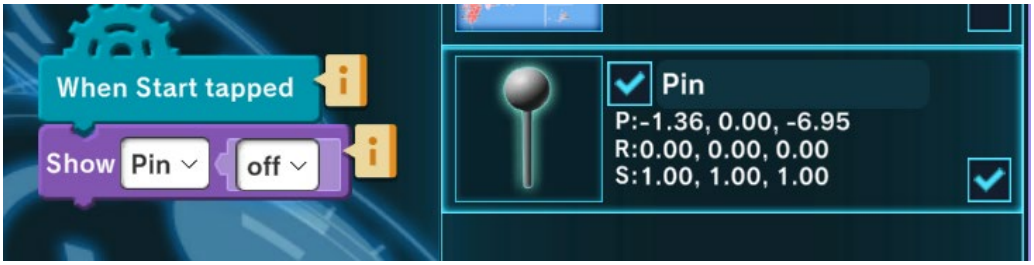

When you run this program, a 3D graph will appear on Gifu prefecture as shown below.




Once a 3D graph is drawn, it will stay unless you delete it with a program. By adding the command block "delete all graphs" as shown in the figure below, you can see the drawing of 3D graphs in animation every time you run the program.




The pins used when identifying coordinates are not needed when drawing the graph, so let's turn off the display of the pins at the start as shown in the following figure.

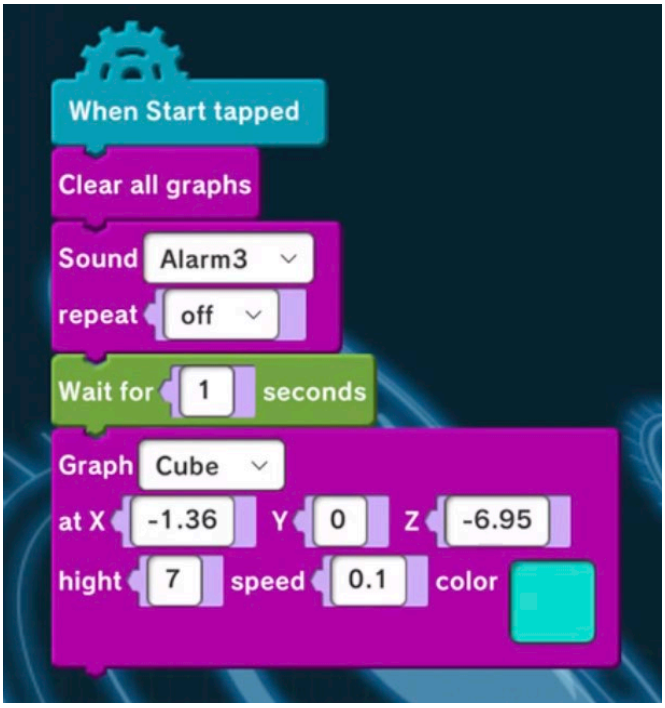



Now you know how to identify the coordinates of any location on a map of Japan and draw a 3D graph anywhere you want. In the next lab, you will draw a graph using statistical data collected on the Internet.

## Advanced Challenge


Play a sound effect when drawing a graph.

You can find many sounds by taping the "Add Object" button and opening the "Sound" tab of the object menu. Tap the play button to listen to the sample sound and select the one you want to use.




The figure on the left shows a sample program that plays a sound. In this example, we use the sound "Alarm3". In this program, the sound is played before drawing the graph, but let's try various patterns to find out when to play the sound to make it more effective.

# Instant noodles consumption

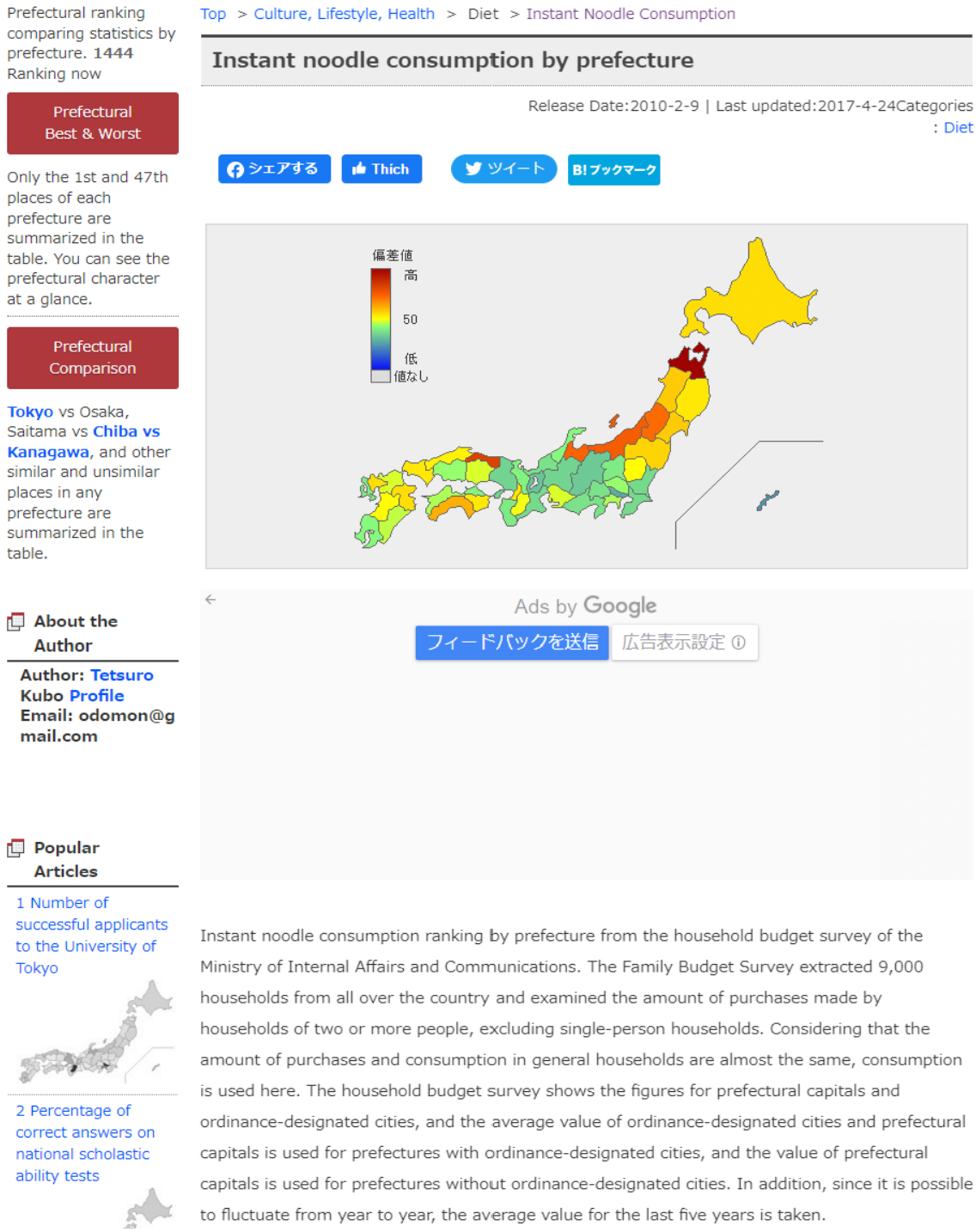
On the Internet, you can find a variety of data to learn about the Japanese people's life. For example, on the website of the Statistics Bureau of the Ministry of Internal Affairs and Communications, you can find all kinds of data such as demographics and business trends. There are also other private websites that compile statistical information.

Statistics Bureau, Ministry of Internal Affairs and Communications: https://www.stat.go.jp


Citizenship seen in statistics and rankings by prefecture: https://todo-ran.com/t/kiji/11811

Here, let’s draw a 3D graph using data from the " “Citizenship seen in statistics and rankings by prefecture”" website.

Use data from "Instant Noodles Consumption by Prefecture". This data can be found on the " “Citizenship seen in statistics and rankings by prefecture” website.



The actual data are shown in the figure below (an excerpt). The actual data are shown in the figure below (an excerpt). The data shown here is the amount of instant noodles purchased by a household in a year, presented in grams (g). This data is based on the results of the 2016 household budget survey conducted by the Statistics Bureau of the Ministry of Internal Affairs and Communications.

☐ **Instant noodle consumption by prefecture**

Click "North/South" and "Descending/Ascending" on the right side of "Alignment" to sort and display.

| rank | prefecture | Consumption | Deviation |
|---|---|---|---|
| Parallel | North South | Descending Ascending | Descending Ascending |
| 1 | Aomori Prefecture | 9,227g | 86.31 |
| 2 | Tottori Prefecture | 8,164g | 73.83 |
| 3 | Niigata Prefecture | 7,742g | 68.87 |
| 4 | Toyama Prefecture | 7,634g | 67.61 |
| 5 | Yamagata Prefecture | 7,523g | 66.30 |
| 6 | Kōchi Prefecture | 6,950g | 59.57 |
| 7 | Miyagi Prefecture | 6,711g | 56.77 |
| 8 | Akita Prefecture | 6,679g | 56.39 |
| 9 | Fukushima-ken | 6,620g | 55.70 |
| 10 | Saga Prefecture | 6,560g | 54.99 |
| 11 | Hokkaido | 6,536g | 54.71 |
| 12 | Oita Prefecture | 6,484g | 54.10 |
| 13 | Tokushima Prefecture | 6,472g | 53.96 |
| 14 | Osaka | 6,438g | 53.56 |
| 15 | Yamaguchi Prefecture | 6,424g | 53.40 |
| 16 | Iwate Prefecture | 6,401g | 53.13 |
| 17 | Shimane Prefecture | 6,315g | 52.12 |
| 18 | Kumamoto Prefecture | 6,232g | 51.14 |
| 19 | Tochigi Prefecture | 6,227g | 51.08 |
| 20 | Nara Prefecture | 6,148g | 50.16 |
| 21 | Aichi Prefecture | 6,102g | 49.62 |

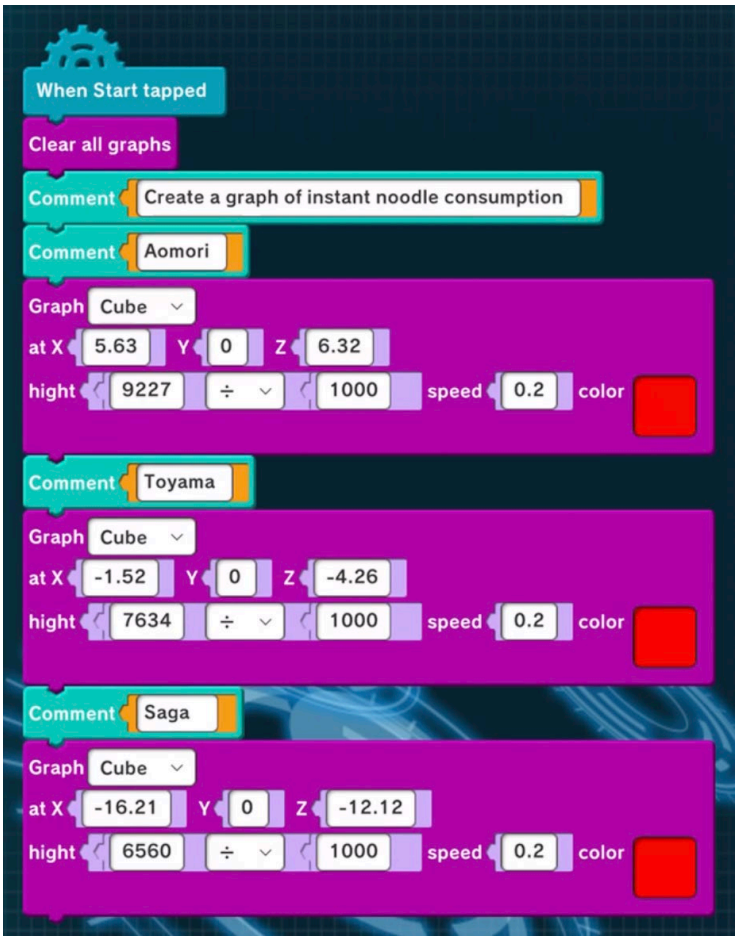
From this data, draw a 3D graph using the following data for Aomori, Toyama, and Saga prefecture.

Aomori Prefecture: 9,227g

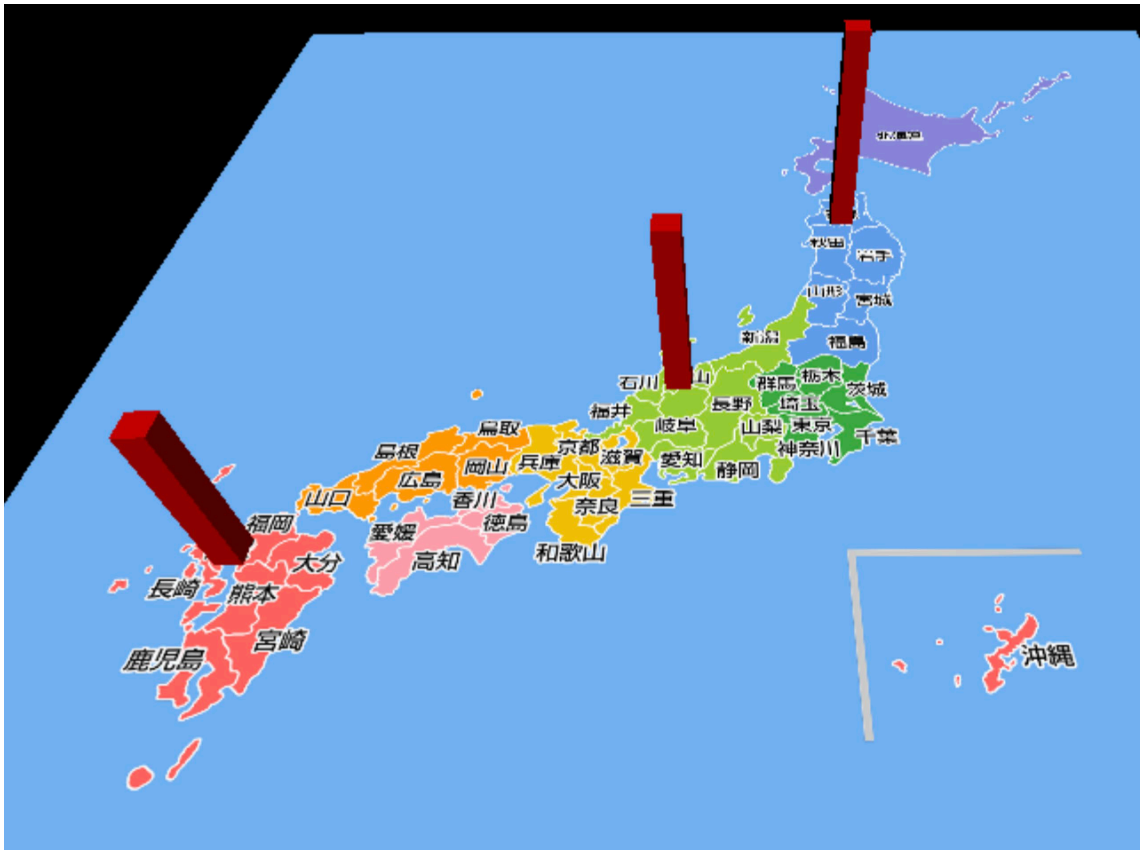
Toyama Prefecture : 7,634g

Saga Prefecture: 6,560g

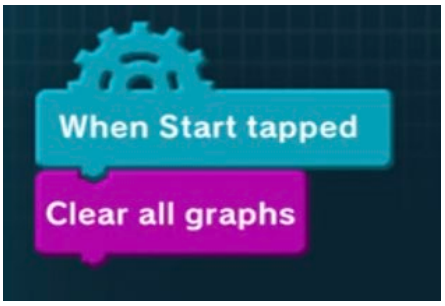
A sample program is provided in Lab 4-3. Select "Japan map" in the object list to display the program as shown below.




When this program runs, 3D graphs animation will be shown on top of the three prefectures as shown in the figure below.



The command in the figure below at the beginning of this program is to delete all 3D graphs that have already been drawn before the program is executed. Without this command, the 3D graphs drawn before execution will remain as it is, which is inconvenient.

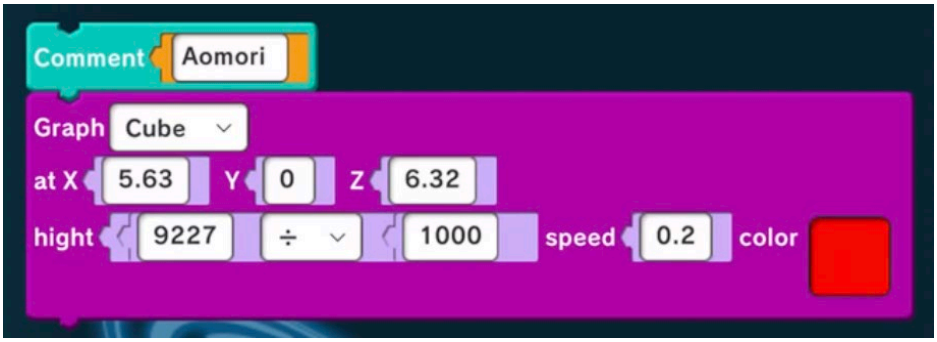



Let's take Aomori prefecture as an example to explain the program itself. In the figure below, where Cube is located is a pull-down menu that allows you to select the type of 3D graph from Cube (square prism) or Cylinder. X, Y, and Z are the coordinates where you draw the graph as seen in Lab 4-2.

The Height of the graph is 9227 (g), which is the amount of instant noodles consumed in Aomori Prefecture, but the number obtained by dividing this value by 1000 is the size of the graph. The data for the other prefectures are also divided by 1000, and the result becomes the size of the graph.

Speed is the speed at which graphs are drawn. The smaller the value, the slower the graph is drawn, and the larger the value, the faster the graph is drawn.

For colors, click the area that shows the color of the square to display the color palette and select your favorite color there.



What should you do if you want to make following changes in the sample program? Please try it.

1）To change the shape of the 3D graph from a Cube to a Cylinder.

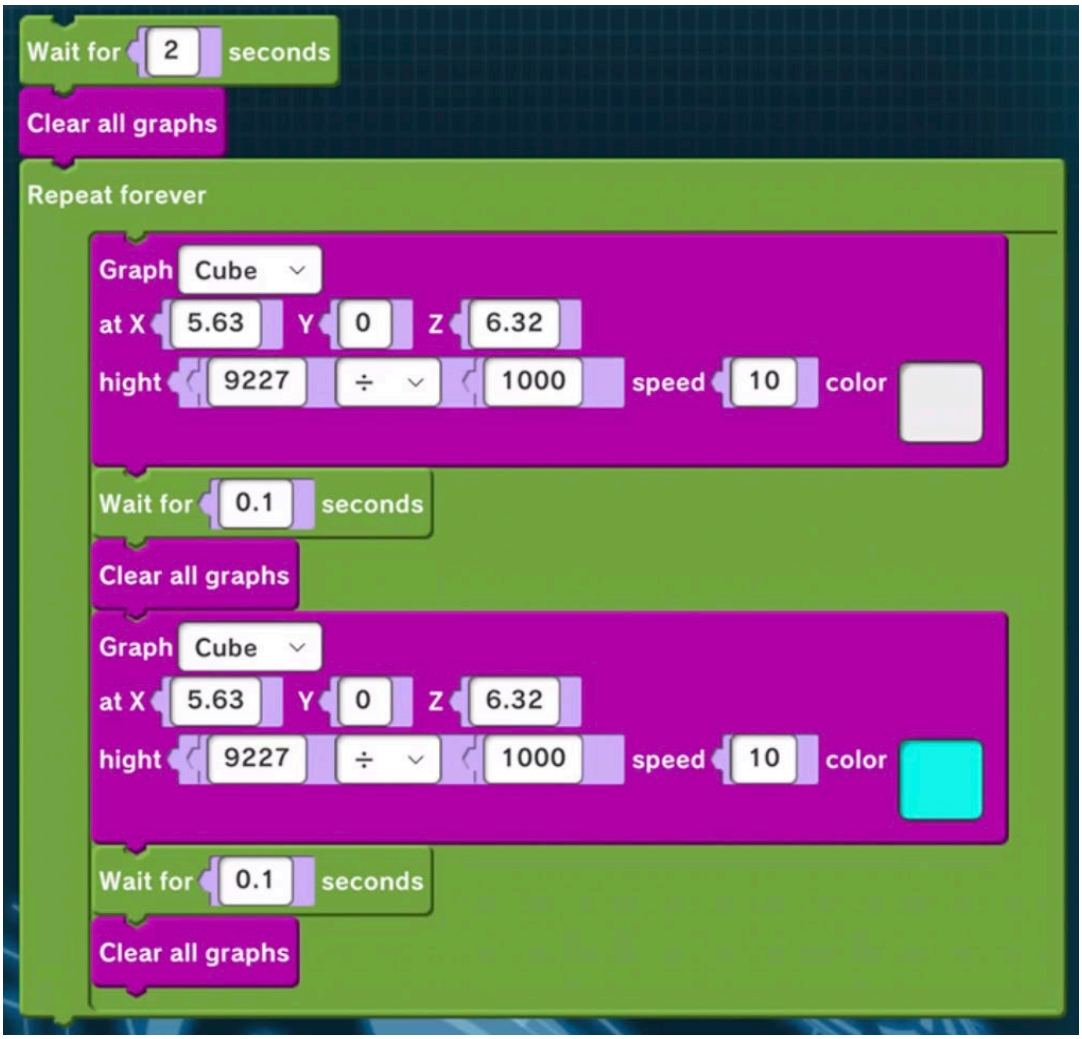
2）To draw a 3D graph quickly.

3）To change the color of the 3D graph to a different color for each of the three prefectures.

Once you become familiar with the 3D graph drawing commands, let's move on to Lab 4-4.

## Advanced Challenge

Change the color of the largest graph in the three prefectures, or blink it to make it stand out. The sample shown below is a program to create a blinking effect by changing color. Add this program below the main sample program in this lab.



# Graph switching

Now you know how to draw a basic 3D graph. In Lab “Graph switching”, you will learn how to show a lot of data while switching among them.

First, let’s prepare the original data. For the original data, select and use three types of data by prefecture from the site below.

https://todo-ran.com
「Citizenship seen in statistics and rankings by prefecture」

The three types of data are as follows:

1）Number of Facebook users
2）Number of convenience stores
3）Population

These three kinds of data are summarized below for three prefectures (Fukushima, Nara, and Kochi). In the table below, coordinates data were also added. Coordinates data were obtained by placing pins.

Statistics by prefecture

| | Column 1 | Column 2 | Column 3 | Column 4 | Column 5 | Column 6 | Column 7 |
|---|---|---|---|---|---|---|---|
| | Prefecture Name | Number of Facebook User | Number of Convenience Store | Population | X coordinate | Y coordinate | Z coordinate |
| Line 1 | Fukushima | 400000 | 889 | 2040000 | 5.1 | 0 | -2.2 |
| Line 2 | Nara | 310000 | 456 | 1399000 | -3.9 | 0 | -9.8 |
| Line 3 | Kochi | 150000 | 286 | 766000 | -9.3 | 0 | -11.6 |

In this lab, make a program to display the data in this table one after another by tapping a button. For this purpose, it is convenient to use the Dataset function provided by Mind Render. Think of a dataset as a container for data with vertical (rows) and horizontal (columns) dimensions. In the table above, for an example, if you want to know the number of convenience stores in Fukushima Prefecture, the number at the intersection of the first row and the third column corresponds to the number. In this way, Dataset can be used to store data organized into rows and columns, which can be read freely from the program.

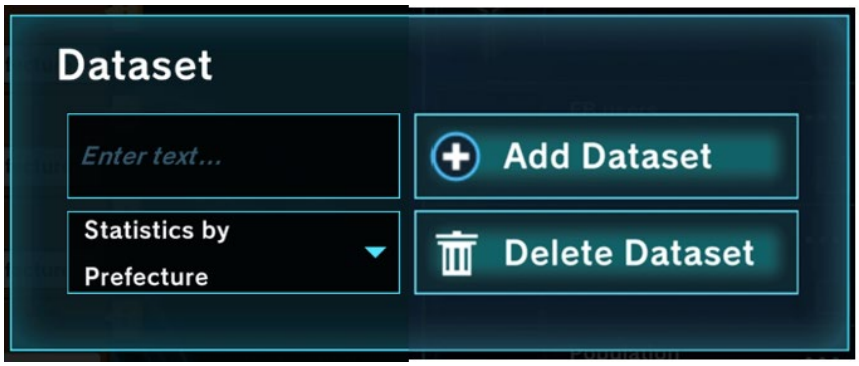
(Example 1) To find the population of Kochi Prefecture -> Row 3, Column 4

(Example 2) To find the coordinates of Nara Prefecture --> X (line 2, column 5), Y (line 2, column 6), Z (line 2, column 7)

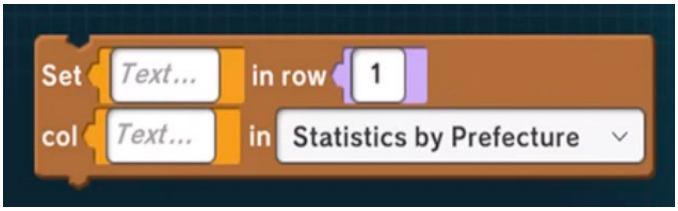
To use Dataset, first create a new dataset as follows.


1) Tap the button labeled “Make Var" in the command category on the left side of the screen.

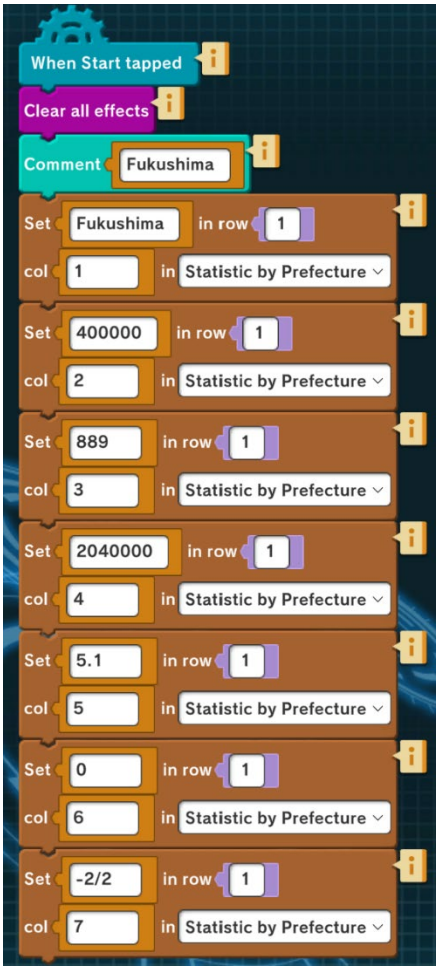
2) Four setting areas are displayed: variables, global variables, lists, and datasets. In the dataset area, enter "Statistics by Prefecture" as shown in the figure below and click the "Add Dataset" button. A new data set named "Statistics by Prefecture" has now been created.



In order to use a dataset, you must first enter the data into the dataset. Tap the "x/y" category to find commands related to the dataset.


To add data into data set, use the command block shown below. Insert data referring to the above "Statistics by Prefecture" table. Be careful not to make a mistake to specify row or column when you add data.




The figure on the left shows the data for Fukushima Prefecture in the first row, columns 1 through 7. In the same way, add the data for Nara and Kochi prefecture. To insert the data of Nara and Kochi prefectures, copy the program of Fukushima prefecture and replace the data with Nara and Kochi prefectures' data to save time. How to copy command block is explained in "Tips for using Mind Render 2 conveniently" at the end of this document.


It would be nice if the person looking at the graphs could use this program and switch buttons to see the stats they wanted to see. Add four buttons to the object list for this purpose. To add buttons, tap "Add Object" button then add it from the Tools tab.

Each button reads specific data (e.g., number of Facebook users) for three prefectures and draws a 3D graph. As an example, look at the program for the number of Facebook users in


the figure below.

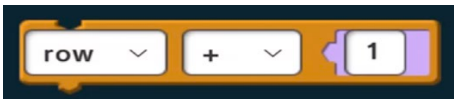
The point of this instruction is that it uses variables to automate data retrieval. The variable "row" is set to 1 as the initial value, and the instructions in the loop are executed. When the variable "row" is set to 1, it will be executed as follows.

Draw a 3D graph with a Cube (spare prism) at the coordinates {X: (Value row 1 col 5 of Statistics by Prefecture), Y: (Value row 1 col 6 of Statistics by Prefecture), Z: (Value row 1

col 7 of Statistics by Prefecture) }. The height of the graph is (Value row 1 col 2 of Statistics by Prefecture) / 100000, the Speed is 0.01, and the Color is white.

You can see what each column in the first row represents by referring to the Statistics by Prefecture table presented earlier. The following is an excerpt of the first column.

| | Column 1 | Column 2 | Column 3 | Column 4 | Column 5 | Column 6 | Column 7 |
|---|---|---|---|---|---|---|---|
| | Prefecture Name | Number of Facebook User | Number of Convenience Store | Population | X coordinate | Y coordinate | Z coordinate |
| Line 1 | Fukushima | 400000 | 889 | 2040000 | 5.1 | 0 | -2.2 |




When you finish drawing the 3D graph, the command on the left will add 1 to the value of the variable "row".

Then the value of the variable "row" is set to 2, the command execution is repeated. Now the program will draw a 3D graph using the data in each column of the second row. The data in the second row will look like below table.

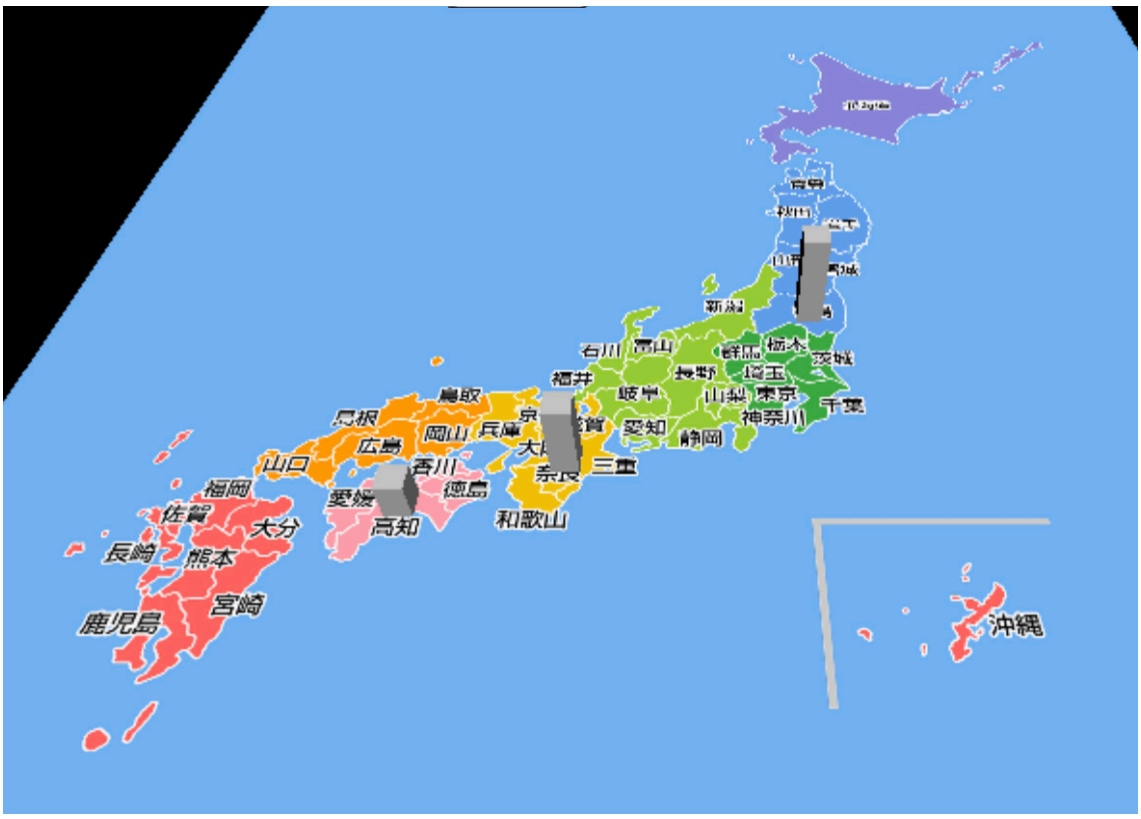
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Line 2 | Nara | 310000 | 456 | 1399000 | -3.9 | 0 | -9.8 |

When you finish drawing a 3D graph with the second row of data, 1 will be added to the variable "row" again.

This time, the value of the variable "row" is set to 3 and the command execution is repeated. This is the third repetition, so this is the last one. Now the program will draw a 3D graph using the data in each column of the third row. The data in the third row will look like below table.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Line 3 | Kochi | 150000 | 286 | 766000 | -9.3 | 0 | -11.6 |

The repetition is now finished, and the program execution is completed. The following figure shows the result of running this program




Looking at the 3D graph, you can see the difference in the number of Facebook users in the three prefectures at a glance.

For the number of convenience stores and population, the basic mechanism of the program is the same as the number of Facebook users. You can save time by copying the program for the number of Facebook users and replacing the numbers.

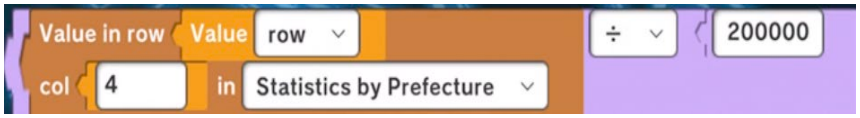
You need to change the number that divides the original data to adjust graph height between 0 and 10. The original data has different ranges of numbers as shown in the figure below.

| Column 2 | Column 3 | Column 4 |
|---|---|---|
| Number of Facebook User | Number of Convenience Store | Population |
| 400000 | 889 | 2040000 |
| 310000 | 456 | 1399000 |
| 150000 | 286 | 766000 |

Therefore, the place to specify the size of the 3D graph will be different as follows。


Number of Facebook users

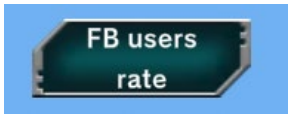

Number of convenience stores




Population

Refer to above three figures, complete the program to draw 3D graphs of the number of convenience stores and population.

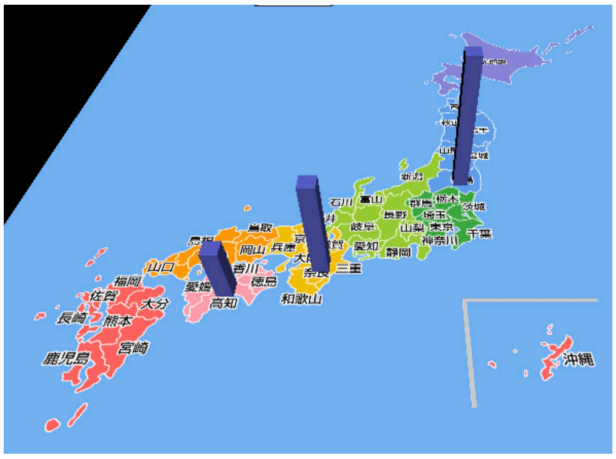
Now, use the remaining "Facebook users' rate" to indicate the number of Facebook users in the three prefectures per 20 prefecture residents.



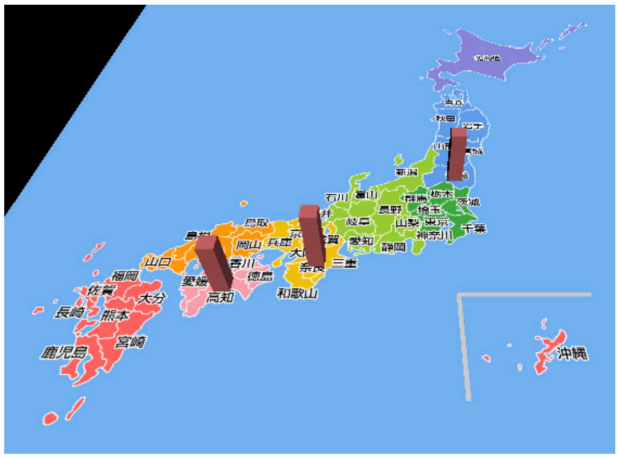

Therefore, you should calculate (number of Facebook users) / (population) as the height of the graph.

Refer to the figure below to make the program. In this example, the calculation result fits in the range of 0-10 for the 3D graph, so there is no need to adjust it.

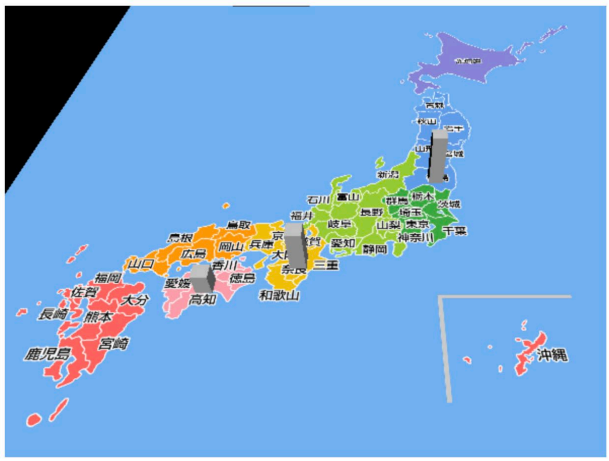

The results of running the program for the four buttons are shown below.



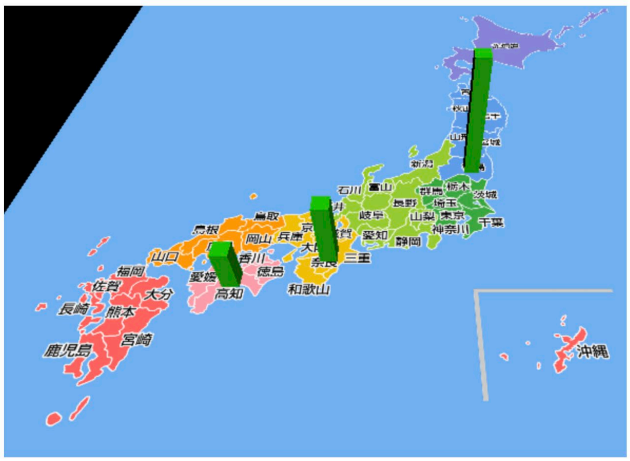
Number of Facebook users



Number of convenience stores




Population




Facebook users’ rate

As can be seen from the two 3D graphs of the number of Facebook users and the percentage of Facebook users, Kochi Prefecture has a smaller number than the other two prefectures in terms of the number of Facebook users, but when looking at Facebook users’ rate, there is not much difference from the other two prefectures. What about the rate of convenience stores? As an exercise, find out the results.

## Advanced Challenge

“Citizenship seen in statistics and rankings by prefecture” （https://todo-ran.com）has a lot of other interesting data. Draw 3D graphs with data you find.

# World population trends

This lab uses a world map instead of a Japan map.

You can find many statistical data of the world on the Internet, but in this lab, data in following website will be used.

Statistics Bureau, Ministry of Internal Affairs and Communications "Statistics of the World 2020"
http://www.stat.go.jp/data/sekai/0116.html

This site provides a variety of data. Let's use "World Population Trends (1950-2050)" as the original data. The table below shows only the original data for 2010. The unit is million people. "North America" refers to the major regions of North America, such as the United States and Canada.

| Area Name | Asia | North America | South America | Europe | Africa | Oceania |
|---|---|---|---|---|---|---|
| 2010 | 4,210 | 343 | 591 | 736 | 1,039 | 37 |

Table: World Population Trends (2010)

We also need the coordinates to draw graphs on the map. Using the method introduced in Lab 4-2 (using pins to identify coordinates), the coordinates of each region are found in the table below.

Table: Summary of coordinates data

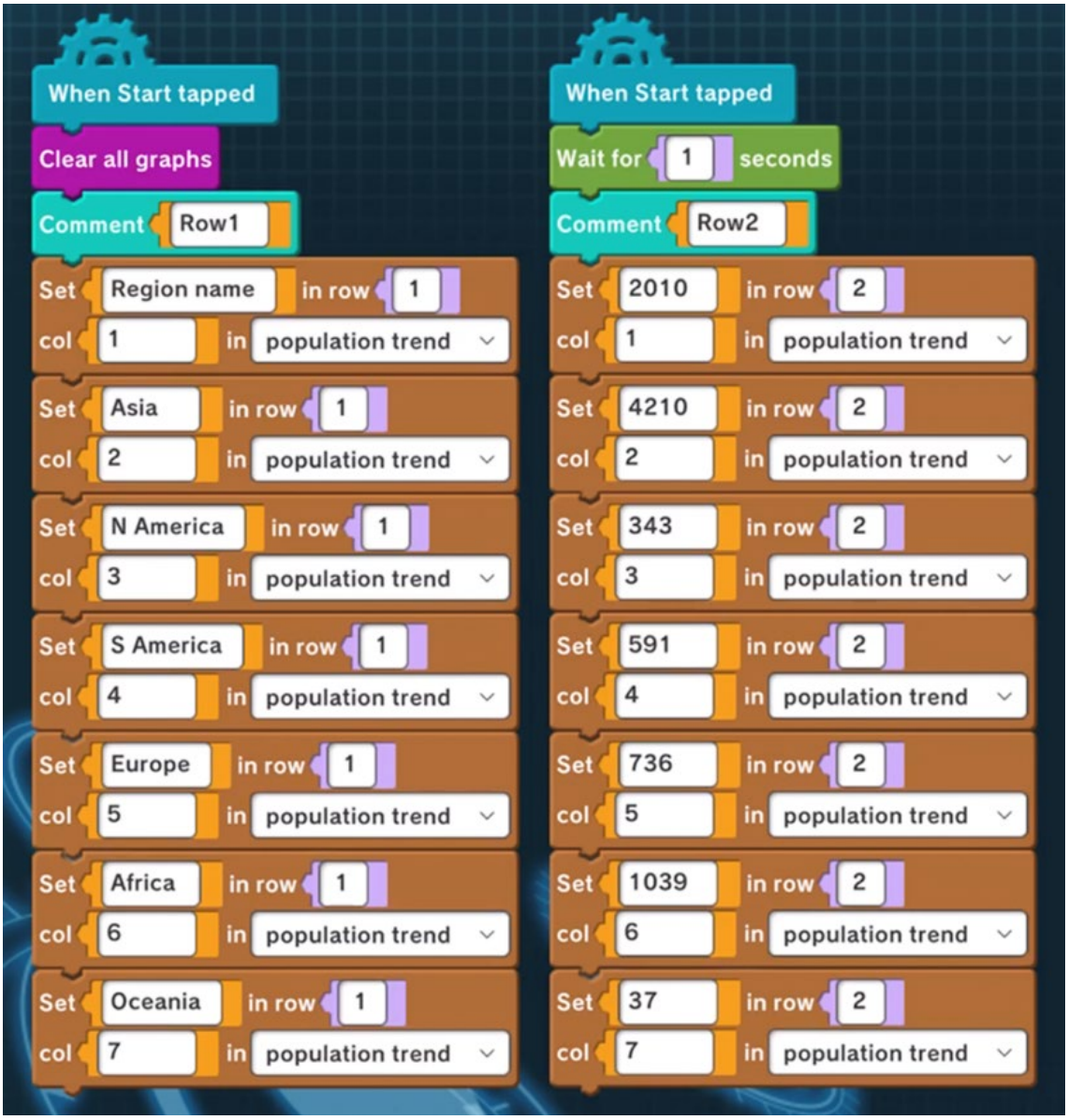
| Area Name | Asia | North America | South America | Europe | Africa | Oceania |
|---|---|---|---|---|---|---|
| x | 20 | -22 | -14 | 2 | 0 | 30 |
| y | 0 | 0 | 0 | 0 | 0 | 0 |
| z | 2 | 4 | -8 | 5 | 0 | -10 |

The program you are going to make will start drawing a graph by taping a button, and by taping another button, the data for each country will be displayed on a graph on the world map one after another.

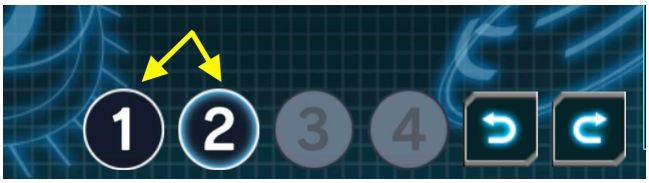
First, enter the data for "World Population Trends". Place the data in a "World map" object.


Set up a new dataset called "Population Trend" in advance.

In order to make it easier to see in the program editor screen, the data is divided into two chunks and each chunk is loaded with a little delay. The reason for having "Clear all graphs" at the beginning is to initialize the screen by deleting the graphs drawn last time.



We will put the coordinate data in the same "world map" object, but the screen will be full and it will be difficult to see, so we will switch to another screen. In Mind Render 4 programming screens can be used per one object. To switch to another screen, tap the arrow button in the figure below.

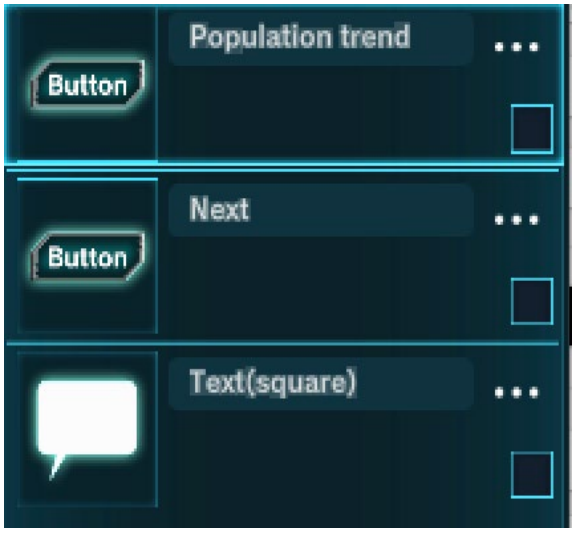



The figure below shows the coordinate data entered on screen ②. Set the dataset with the name "Coordinates".


The following functions are incorporated for convenient operation for users.

- Tap a button to start drawing a graph
- Tap another button to display a graph for the next data
- Display the values of the graph in numbers

Add the following three objects for the above three functions.



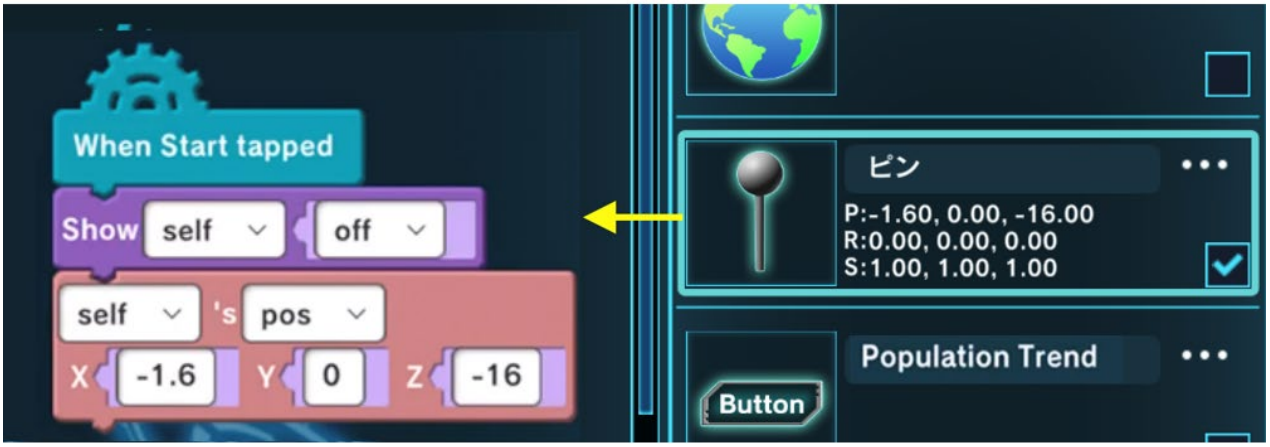

Tap this button to draw a graph using the world Population Trends data.

Tap this button to draw a graph using the next data.

The values of the graph will be displayed in this Text(square).

The button and Text(square) are placed as shown below. In the image below, the black square above Africa is the Text(square).

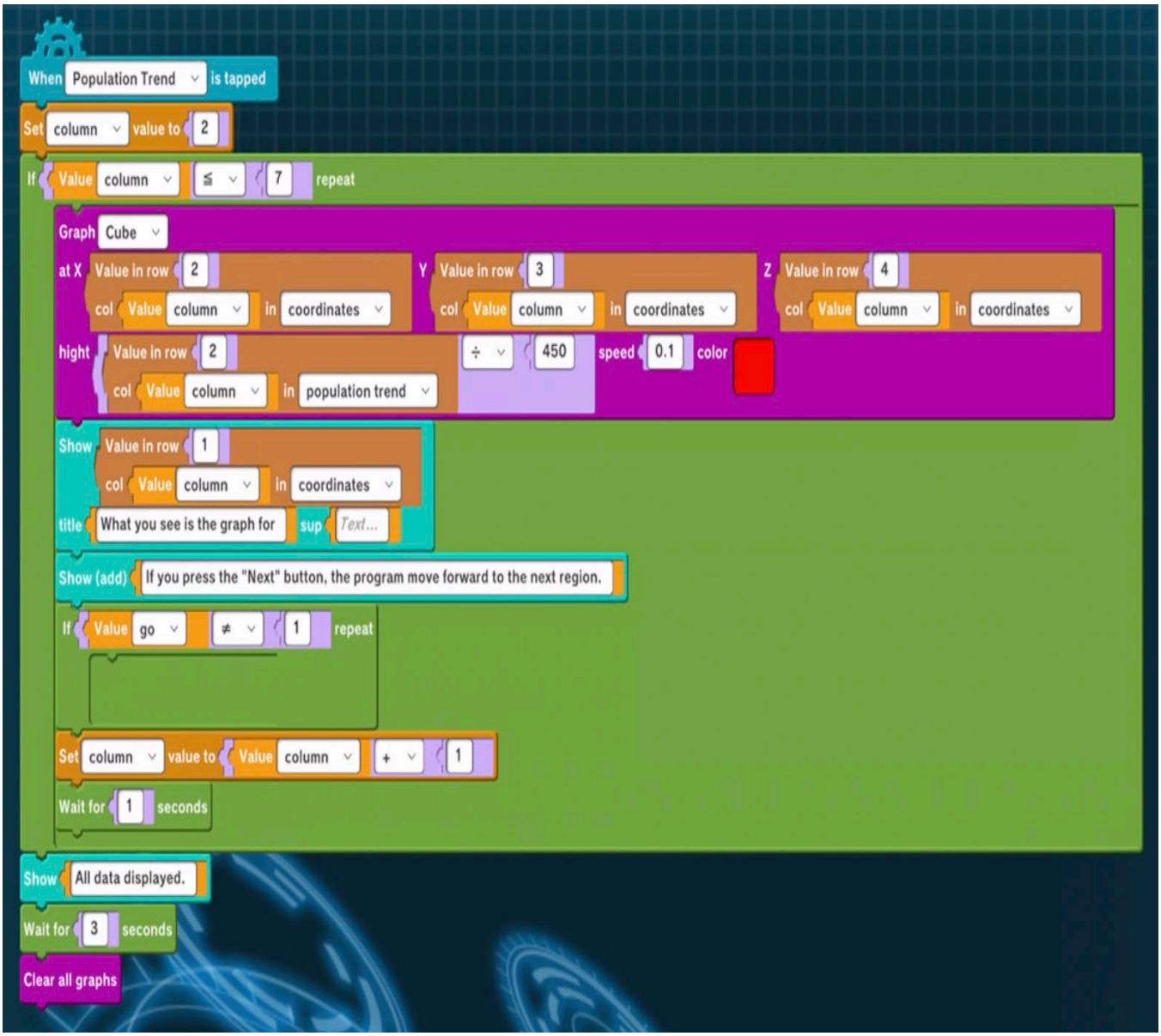
Move the pin you used to find the coordinates on the map out of the way and turn off the display.





The program to draw the graph will be created in the "Population Trend" button.

The following figure shows the entire program. It is divided into several parts and explain below.



The entire program runs when the "Population Trend" button is tapped. When executed, at first the initial values of two variables, "row" and "column", are set to 2. Of course, these two variables themselves need to be set up in advance. (Please refer to Lab “Drone game” for how to set up a variable.)


The reason we set the initial value of "column" to 2 is because the first column of the two original data tables is the heading column. In the figure below, you can see that in both of the two tables, the second column is where the first data is placed.

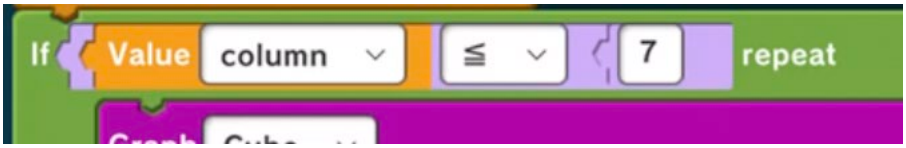
Line 2

| Area Name | Asia | North America | South America | Europe | Africa | Oceania |
|---|---|---|---|---|---|---|
| 2010 | 4,210 | 343 | 591 | 736 | 1,039 | 37 |

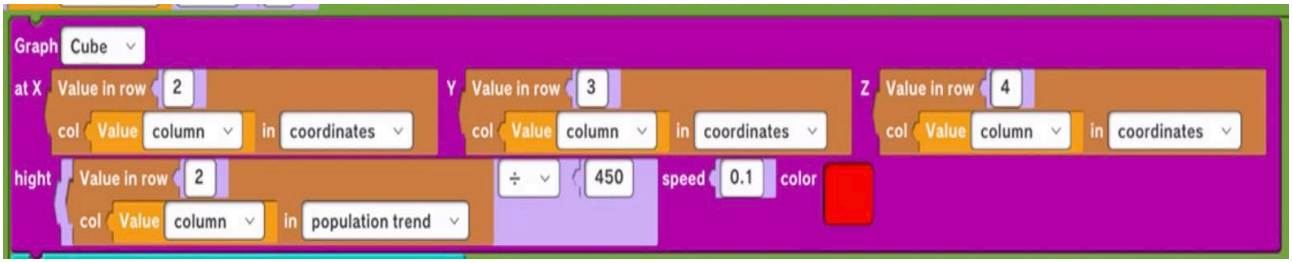
Line 2

| Area Name | Asia | North America | South America | Europe | Africa | Oceania |
|---|---|---|---|---|---|---|
| x | 20 | -22 | -14 | 2 | 0 | 30 |
| y | 0 | 0 | 0 | 0 | 0 | 0 |
| z | 2 | 4 | -8 | 5 | 0 | -10 |

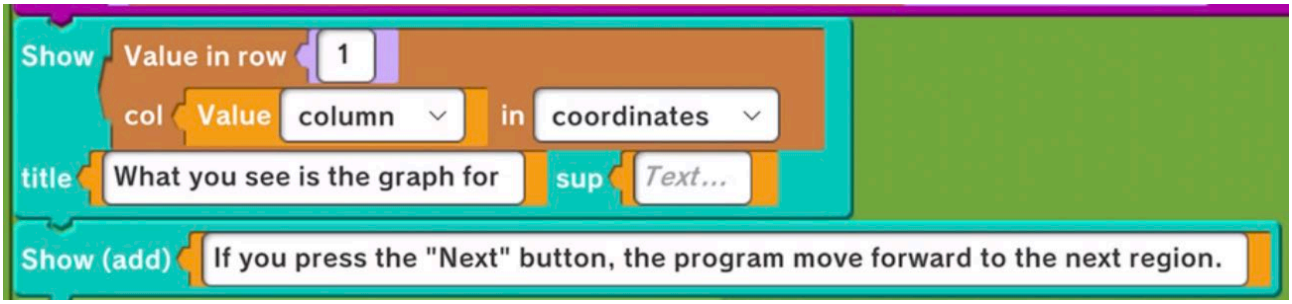
Since the command for drawing the graph should be the same for Asia in column 2 through Oceania in column 7, repeat the same command until the variable "column" becomes 7.




The program for drawing the graph is shown in the figure below. For X, Y, and Z coordinates, data is read from the second to fourth rows of the column specified by the variable "column" of the "coordinates" data set. The data that determines the height of the graph is read from the second row of the "Population Trend" data set. Divide the read numbers by an appropriate number to fit within the range of minimum value 0 to maximum value 10 for graph drawing. Finding this number is a matter of trial and error, trying different numbers so that the difference in graph size is noticeable and the largest data does not exceed the maximum size of the graph (10). Here we divide by 450 which we found by doing like that.

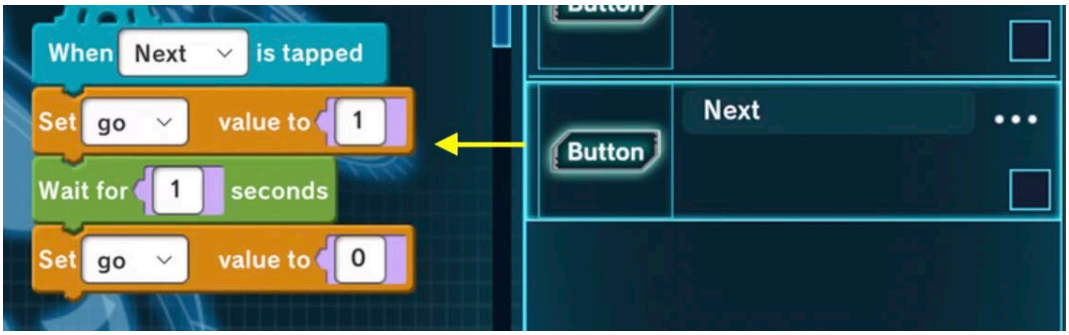



After the graph is drawn, it is displayed on the screen by using a number command to display "What you see is the graph for (region name)" on the screen so that it is easy for the user to understand. The first row of the "coordinates" dataset is used for (region name).

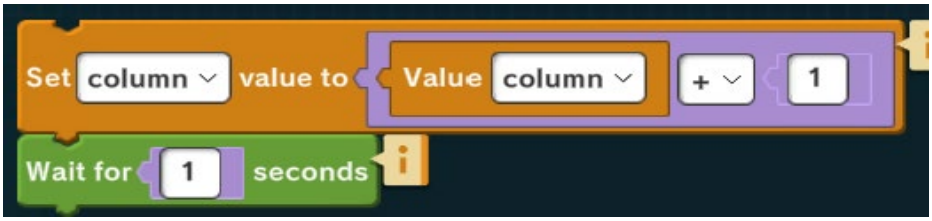


After each region is displayed, it does not move forward until the "Next" button is tapped to allow time to see the size of the graph and the actual numbers. The variable "Go" is set up in advance, until the value of this variable is set to 1, the program does not do anything but repeating. When the value of the variable "Go" becomes 1, it waits 1 second, then increases the column value by 1 and loops back to the beginning of the inner nested repetition.


The program to change the value of the variable Go" is located in the "Next" button. When the button is tapped, it sets the value of the variable "Go" to 1. After one second wait, it immediately sets the value back to 0.

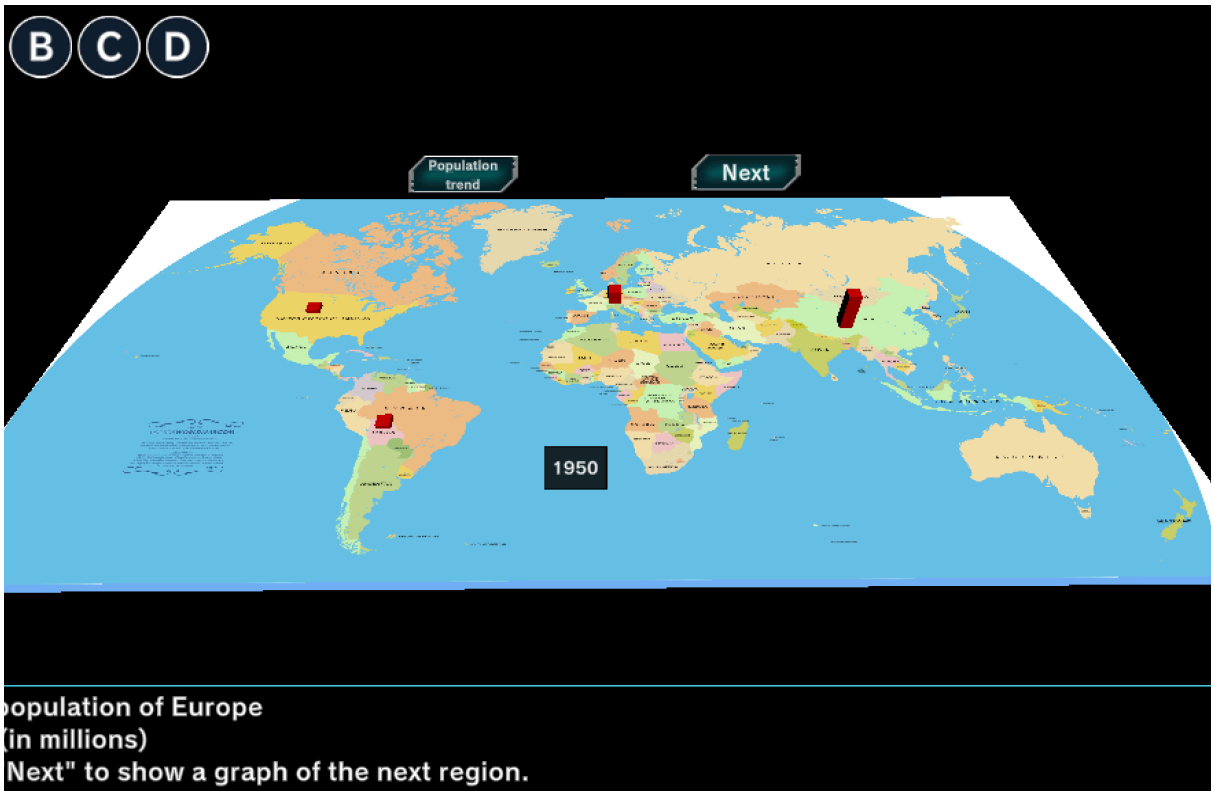



After one repetition, it goes back to the beginning of the repetition and runs with the data from the next column. Before repeating, the previously drawn graphs will be deleted.



When all the graphs are displayed, below message is displayed. Then it waits for 3 seconds before deleting all the graphs for the new session.。


When you run the program, it will look like the figure below.




You can find a lot of interesting data on world statistics on the Internet. Draw graphs on the world map with the data you find.

## Advanced Challenge

Let's extend the program to show the data for each region from 1950 to 2010 in graphs

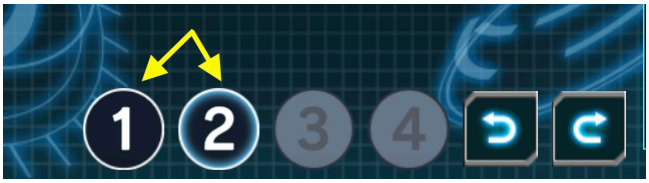
| Area Name | Asia | North America | South America | Europe | Africa | Oceania |
|---|---|---|---|---|---|---|
| 1950 | 1,405 | 173 | 169 | 549 | 228 | 13 |
| 1960 | 1,705 | 205 | 220 | 605 | 283 | 16 |
| 1970 | 2,142 | 231 | 287 | 657 | 363 | 20 |
| 1980 | 2,650 | 254 | 361 | 694 | 476 | 23 |
| 1990 | 3,226 | 280 | 443 | 721 | 630 | 27 |
| 2000 | 3,741 | 312 | 522 | 726 | 811 | 31 |
| 2010 | 4,210 | 343 | 591 | 736 | 1,039 | 37 |

Table : World Population Trends（1950～2010）


The “World Population Trends" program is placed in the "World map" object.

Since there is a large amount of data, they are divided into 8 chunks and each chunk is loaded at a slightly different time.

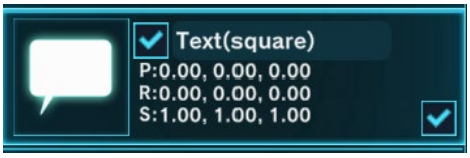
```
When Start tapped
Clear all graphs
Comment Row1
Set Region name in row 1 col 1 in population trend
Set Asia in row 1 col 2 in population trend
Set N America in row 1 col 3 in population trend
Set S America in row 1 col 4 in population trend
Set Europe in row 1 col 5 in population trend
Set Africa in row 1 col 6 in population trend
Set Oceania in row 1 col 7 in population trend

When Start tapped
Wait for 0.1 seconds
Comment Row2 : 1950
Set 1950 in row 2 col 1 in population trend
Set 1405 in row 2 col 2 in population trend
Set 173 in row 2 col 3 in population trend
Set 169 in row 2 col 4 in population trend
Set 549 in row 2 col 5 in population trend
Set 228 in row 2 col 6 in population trend
Set 13 in row 2 col 7 in population trend

When Start tapped
Wait for 0.2 seconds
Comment 1960
Set 1960 in row 2 col 1 in population trend
Set 1705 in row 2 col 2 in population trend
Set 205 in row 2 col 3 in population trend
Set 220 in row 2 col 4 in population trend
Set 605 in row 2 col 5 in population trend
Set 283 in row 2 col 6 in population trend
Set 16 in row 2 col 7 in population trend

When Start tapped
Wait for 0.3 seconds
Comment 1970
Set 1970 in row 2 col 1 in population trend
Set 2142 in row 2 col 2 in population trend
Set 231 in row 2 col 3 in population trend
Set 287 in row 2 col 4 in population trend
Set 657 in row 2 col 5 in population trend
Set 363 in row 2 col 6 in population trend
Set 20 in row 2 col 7 in population trend
```




The coordinate data will be created on screen ②.

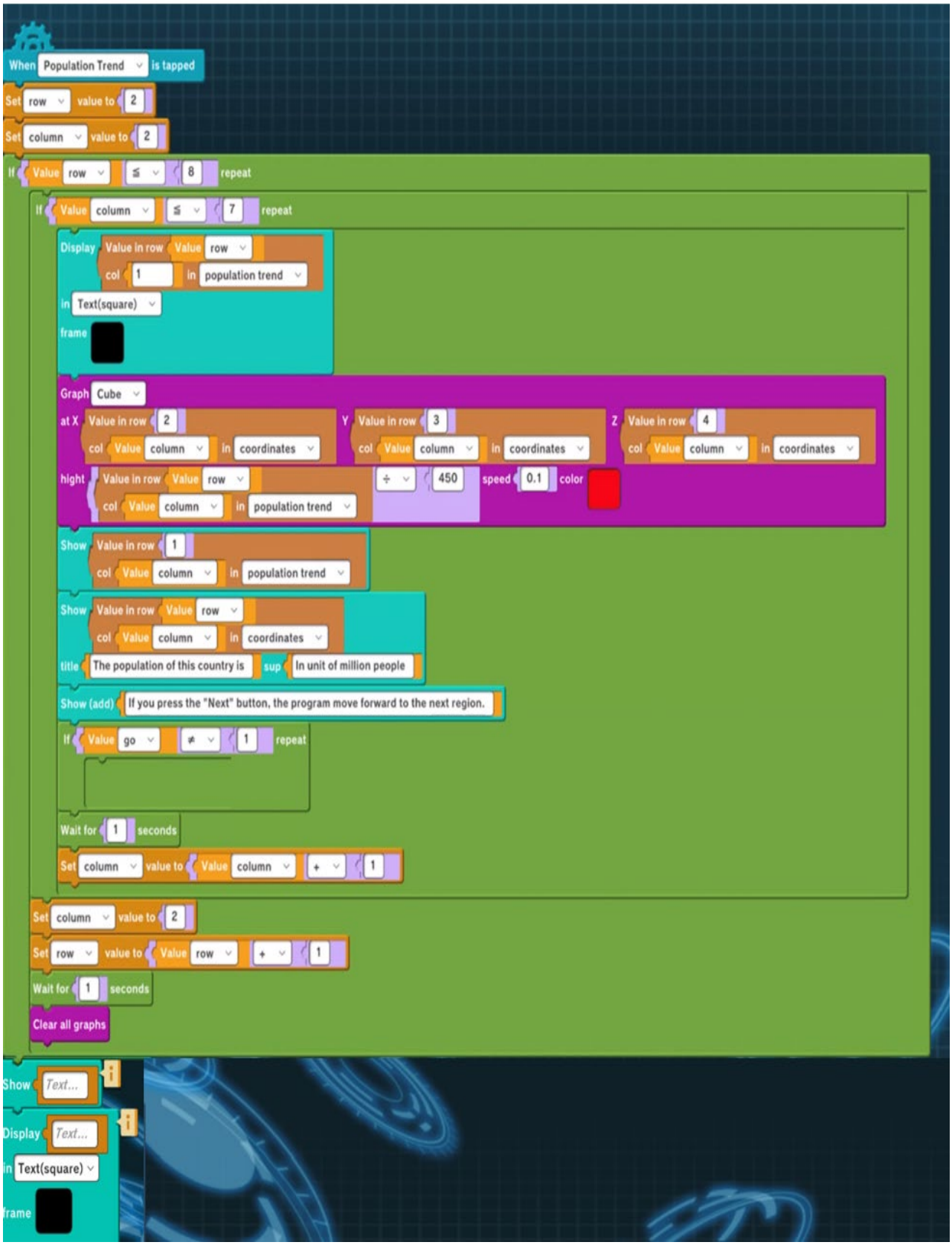
```
When Start tapped
Wait for 0.4 seconds
Comment 1980
Set 1980 in row 2 col 1 in population trend
Set 2650 in row 2 col 2 in population trend
Set 254 in row 2 col 3 in population trend
Set 361 in row 2 col 4 in population trend
Set 694 in row 2 col 5 in population trend
Set 476 in row 2 col 6 in population trend
Set 23 in row 2 col 7 in population trend

When Start tapped
Wait for 0.5 seconds
Comment 1990
Set 1990 in row 2 col 1 in population trend
Set 3226 in row 2 col 2 in population trend
Set 280 in row 2 col 3 in population trend
Set 443 in row 2 col 4 in population trend
Set 721 in row 2 col 5 in population trend
Set 630 in row 2 col 6 in population trend
Set 27 in row 2 col 7 in population trend

When Start tapped
Wait for 0.6 seconds
Comment 2000
Set 2000 in row 2 col 1 in population trend
Set 3741 in row 2 col 2 in population trend
Set 312 in row 2 col 3 in population trend
Set 522 in row 2 col 4 in population trend
Set 726 in row 2 col 5 in population trend
Set 811 in row 2 col 6 in population trend
Set 31 in row 2 col 7 in population trend

When Start tapped
Wait for 0.7 seconds
Comment 2010
Set 2010 in row 2 col 1 in population trend
Set 4210 in row 2 col 2 in population trend
Set 343 in row 2 col 3 in population trend
Set 591 in row 2 col 4 in population trend
Set 736 in row 2 col 5 in population trend
Set 1039 in row 2 col 6 in population trend
Set 37 in row 2 col 7 in population trend
```


In this advanced challenge, the original numerical values of the graph will be displayed. In addition, the year of data for the currently displayed graph will be displayed somewhere on the screen. For this purpose, an object called "Text(square)" will be added in addition to the "Population trend" button and the "Next" button used in Lab 4-5.

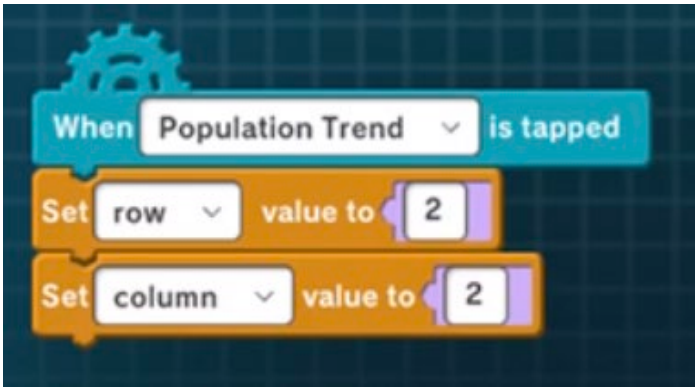


Place the buttons and text rectangle as shown below.

The following figure shows the entire program. It is divided into several parts and explain below.



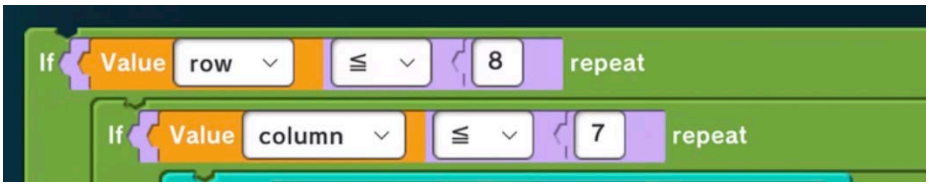
The entire program start running when the "Population Trend" button is tapped. When executed, it first sets the initial values of the two variables, "row" and “column" to 2.
Of course, these two variables need to be set up in advance.




The reason we set the initial value of "column" to 2 is because the first column of the two original data tables is the heading column. In the figure below, you can see that in both of the two tables, the second column is where the first data is placed.

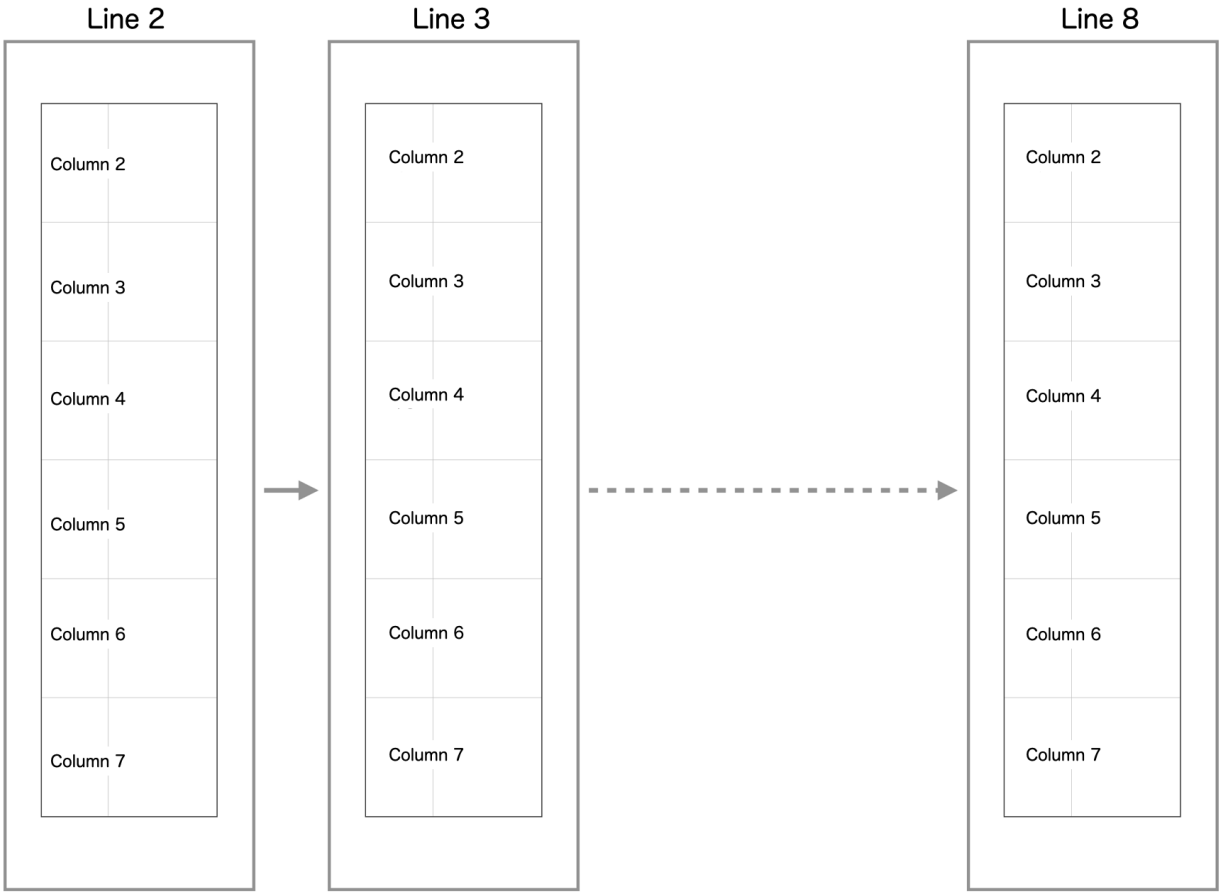
| Area Name | Asia | North America | South America | Europe | Africa | Oceania |
|---|---|---|---|---|---|---|
| 1950 | 1,405 | 173 | 169 | 549 | 228 | 13 |
| 1960 | 1,705 | 205 | 220 | 605 | 283 | 16 |
| 1970 | 2,142 | 231 | 287 | 657 | 363 | 20 |
| 1980 | 2,650 | 254 | 361 | 694 | 476 | 23 |
| 1990 | 3,226 | 280 | 443 | 721 | 630 | 27 |
| 2000 | 3,741 | 312 | 522 | 726 | 811 | 31 |
| 2010 | 4,210 | 343 | 591 | 736 | 1,039 | 37 |

| Area Name | Asia | North America | South America | Europe | Africa | Oceania |
|---|---|---|---|---|---|---|
| x | 20 | -22 | -14 | 2 | 0 | 30 |
| y | 0 | 0 | 0 | 0 | 0 | 0 |
| z | 2 | 4 | -8 | 5 | 0 | -10 |




Next, here is a double nested repetition.

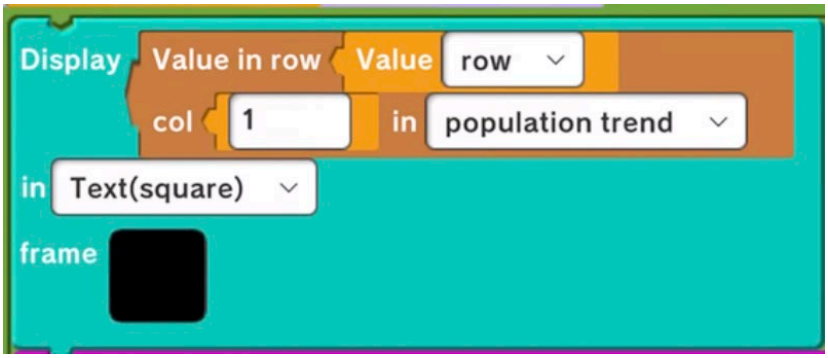
Since the initial value of both "row" and "column" is 2, the "column" value will repeat from 2 to 7, and the "row" value will repeat from 2 to 8. This nested repetition begins with the initial value of 2 for "row" and repeats until "column" starts from 2 and reaches 7. When the "column" exceeds 7, the next "row" value becomes 3 next, and the column repeats from 2 to 7 again. This is repeated until the number of "rows" reaches 8. The figure below shows this.




By running the nested repetitions as shown in the figure above, you can read all the data from row2, column2 to the end of row 8, column 7 included in the “World Population Trends” data. See the figure below.

| Area Name | Asia | North America | South America | Europe | Africa | Oceania |
|---|---|---|---|---|---|---|
| 1950 | 1,405 | 173 | 169 | 549 | 228 | 13 |
| 1960 | 1,705 | 205 | 220 | 605 | 283 | 16 |
| 1970 | 2,142 | 231 | 287 | 657 | 363 | 20 |
| 1980 | 2,650 | 254 | 361 | 694 | 476 | 23 |
| 1990 | 3,226 | 280 | 443 | 721 | 630 | 27 |
| 2000 | 3,741 | 312 | 522 | 726 | 811 | 31 |
| 2010 | 4,210 | 343 | 591 | 736 | 1,039 | 37 |

The program in the figure below at the beginning of the repetitive process reads the first column of each row and display it in the "Text(square)" object. The first column of each row is the year data, as you can see in the above figure. It is a means to show that the graph to be displayed from now on is the data of "XXXX" year before displaying the graph.




Next is the program for drawing the graphs. The x, y, z coordinates for each region to draw graphs are read from the "Coordinate" dataset. If you compare the "World Population Trends" table with the "Coordinates data" table, you will see that in both data, columns 2 through 7 correspond to each region (Asia, North America, etc.). For example, to read the x, y, and z coordinates of Asia, you can read the 2nd row 2nd row, 3rd row 2nd column, and 4th row 2nd column as shown in the figure below. Similarly, for North America a, read 2nd row 3rd column, 3rd row 3rd column, 4th row 3rd column.

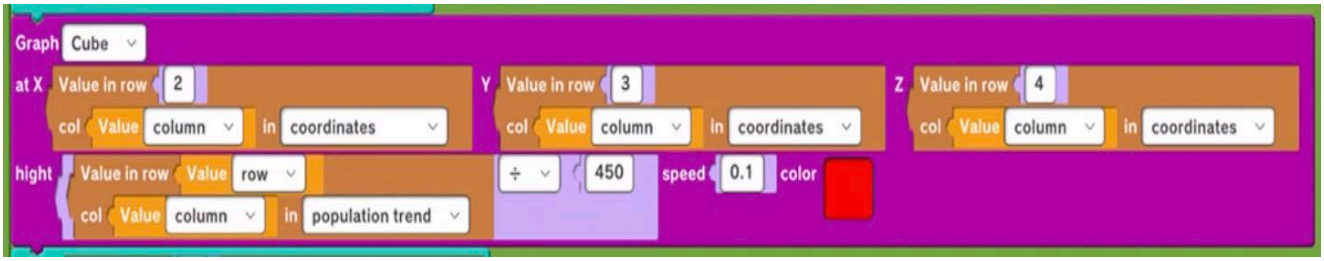
| Area Name | Asia | North America | South America | Europe | Africa | Oceania |
|---|---|---|---|---|---|---|
| x | 20 | -22 | -14 | 2 | 0 | 30 |
| y | 0 | 0 | 0 | 0 | 0 | 0 |
| z | 2 | 4 | -8 | 5 | 0 | -10 |

Line 2/Column 2 - Line 4/Column 2

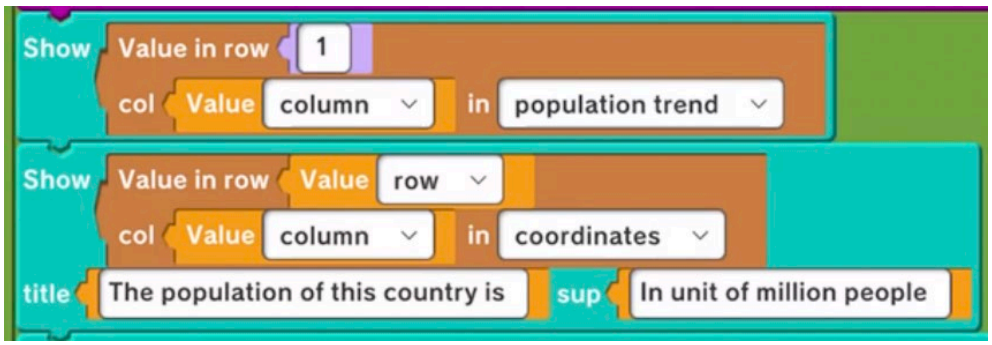
Line 2/Column 3 - Line 4/Column 3

Line 2/Column 7 - Line 4/Column 7

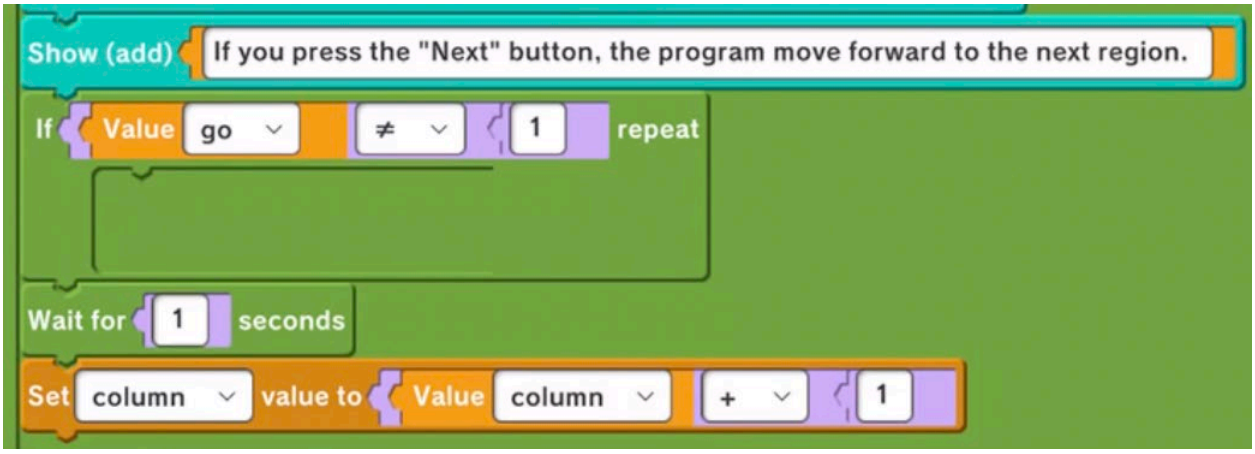
The program for drawing the graph is shown in the figure below. The data for the height of the graph is read from the “Population Trend" dataset. In the example below, the initial values of the variables "row" = 2 and "column" = 2 are used to read the data in the 2nd row and 2nd column of the "Population Trend" dataset. The read number is divided by 450 to fit within the range of minimum value 0 to maximum value 10 for graph drawing. Please refer to the explanation of the lab "World Population" for why it is divided by 450.



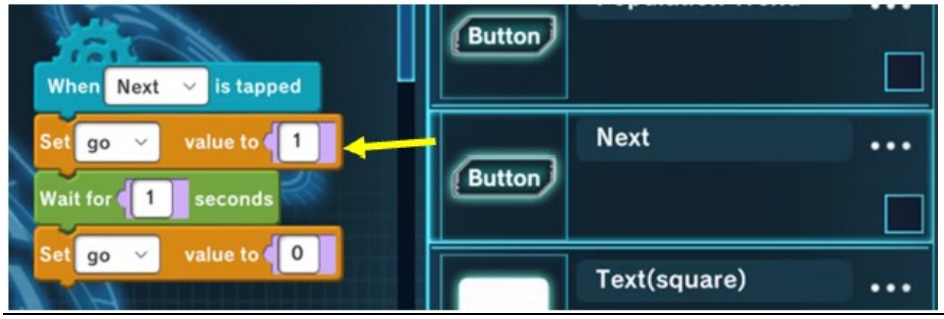
The program in the following figure is to display the original data of the graph in numerical values. For example, if the initial values "row" = 2 and "column" = 2, "The population of Asia is 1405 (in unit of million people)" is displayed in the area at the bottom of the screen.



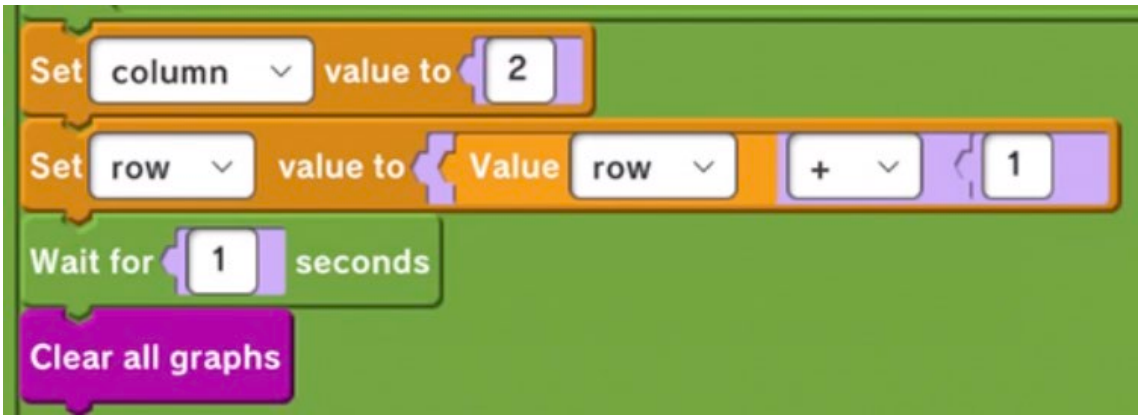

The final part of the repetition inside the nest is shown below. After each region is displayed, the program does not move forward until the "Next" button is tapped to allow time to see the size of the graph and the actual numbers. The variable "Go" is set up in advance, until the value of this variable is set to 1, the program does not do anything but repeating. When the value of the variable "Go" becomes 1, it waits 1 second, then increases the column value by 1 and loops back to the beginning of the inner nested repetition.



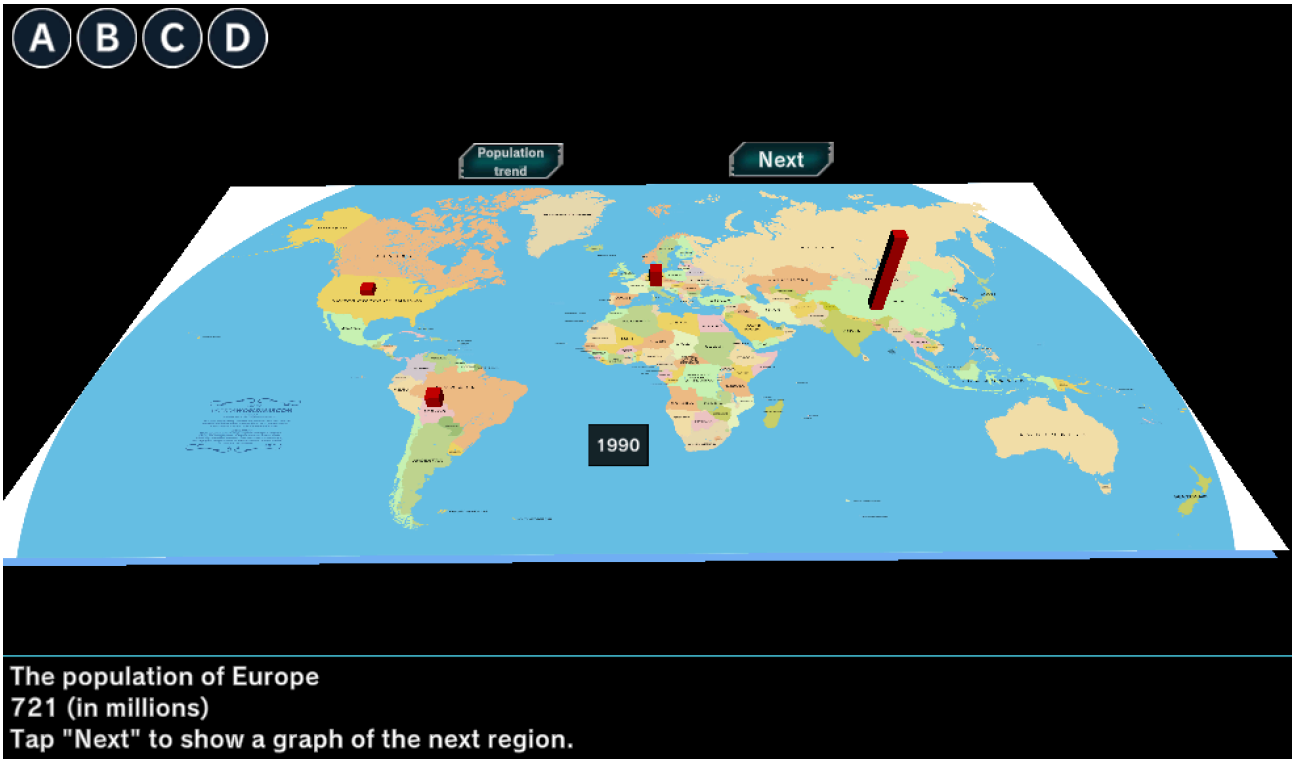

The program to change the value of the variable Go" is located in the "Next" button. When the button is tapped, it sets the value of the variable "Go" to 1. After one second wait, it immediately sets the value back to 0.



When the inner nested repetition completes, it loops back to the beginning of the outer repetition and repeats with the next row of data. When the initial value of the variable "row" = 2, the population data of each region in 1950 was shown in a graph. At the second round of the outer repetition, the value of the variable "row" is 3, and graphs of the 1960 population data for each region is drawn. Delete the graph drawn so far before doing the outer repetition.




When you run the program, it will look like the figure below.




You can find a lot of interesting data on world statistics on the Internet. Draw graphs on the world map with the data you find.

# 10. Let’s study a little 3: "Run a Circuit"

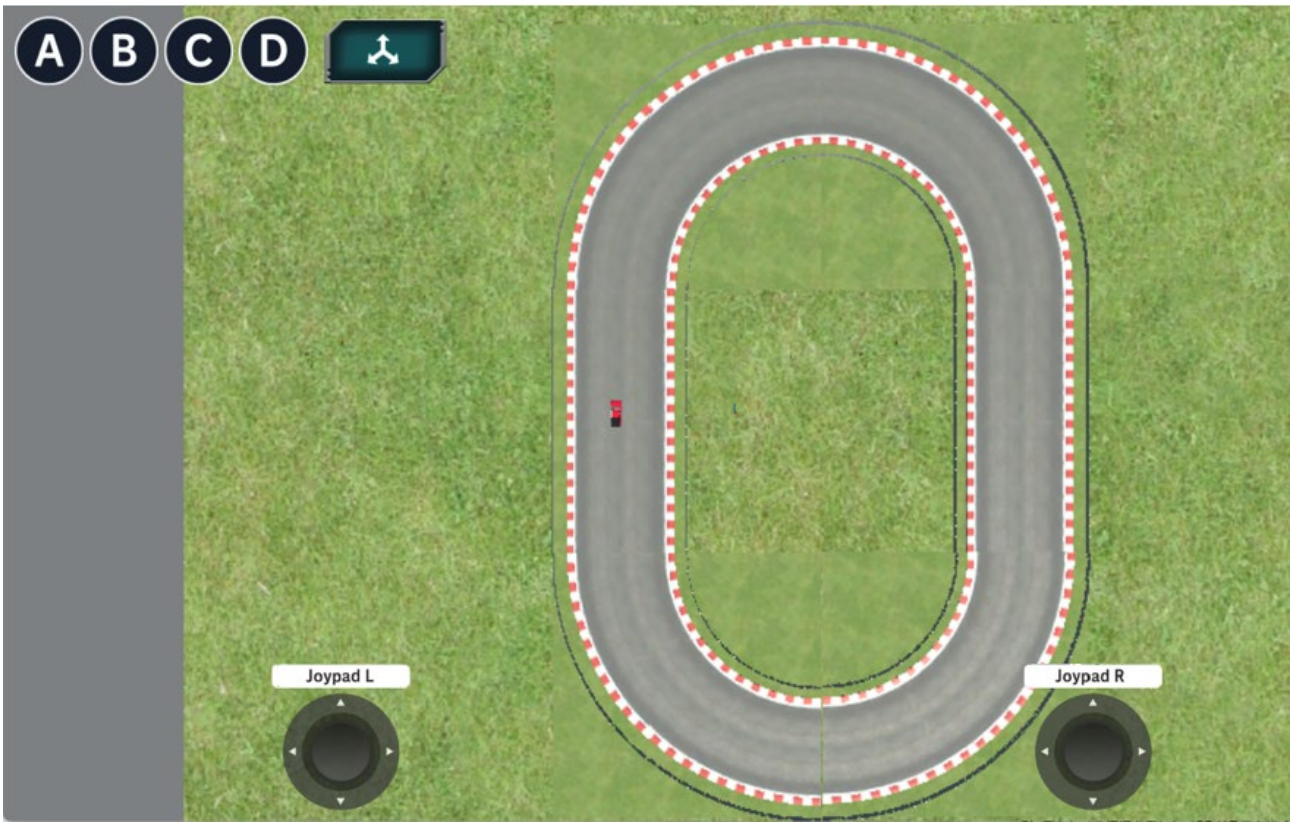
## Run a Circuit

In Lab "Run a Circuit," you will drive a race car. You can use the joypads to control the car, or record the movements of the car, or challenge to program the car to run automatically.

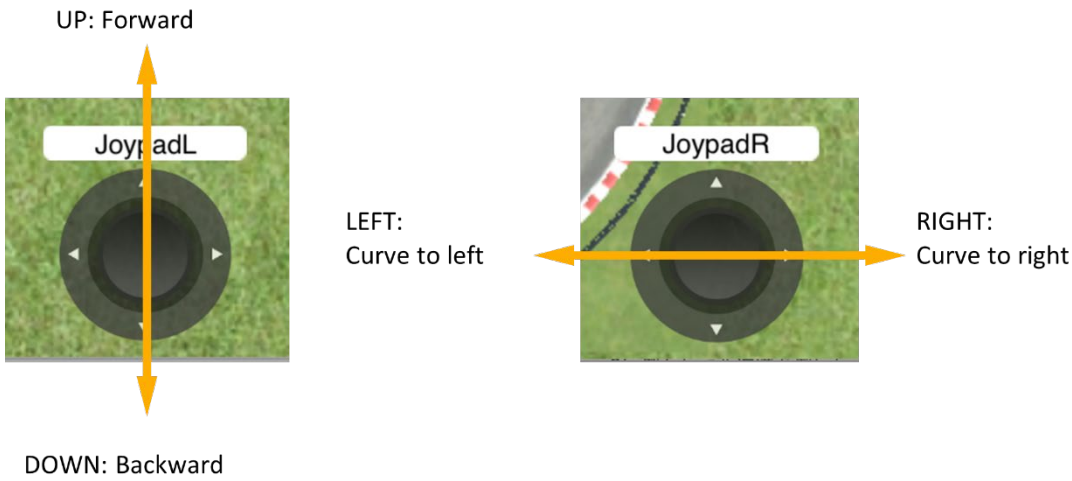
Let's start by selecting "Let's study a little" from the lab menu, then select "Car operation" in the lab "Run a circuit".

# Lab ”Car operation”

Once Lab “Car operation” is opened, change the screen mode to full screen. You can see a red car on the left side of the oval circuit track. In Lab “Car operation”, you will drive this car around the track using the two joypads at the bottom of the screen. Let's run the program by tapping the Start button.



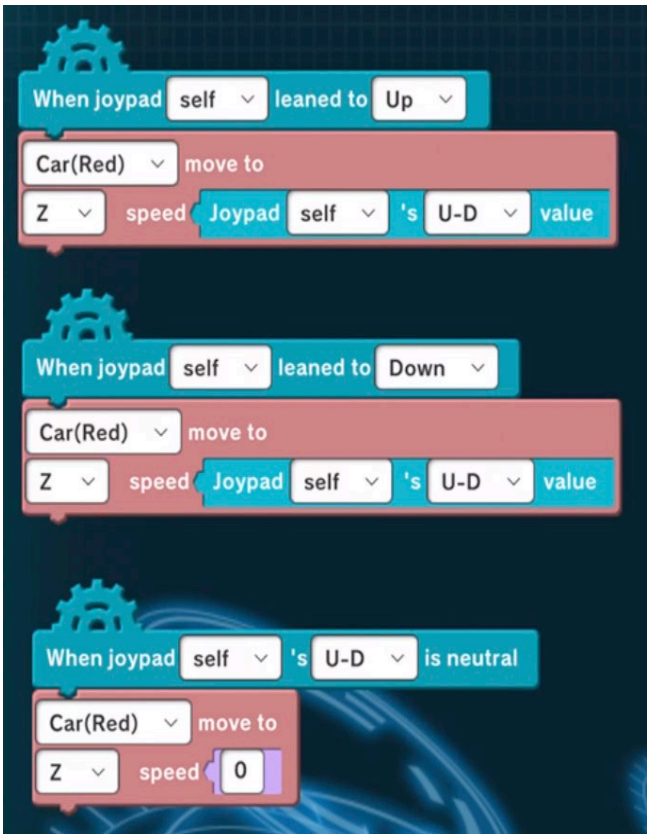

The usage of the two joypads is shown below.



It is difficult to operate two joypads at once on a computer, so, if possible, try using a tablet or a computer with a touch screen.

It may be difficult to operate at first, but you will soon get used to it after a few laps around the circuit. Practice your car maneuvers by testing how fast you can enter a curve and how much you can change the direction of the car.

Once you are comfortable with the control, you can explore the program in Lab “Car operation”. The program for Joypad L is shown in the figure below.

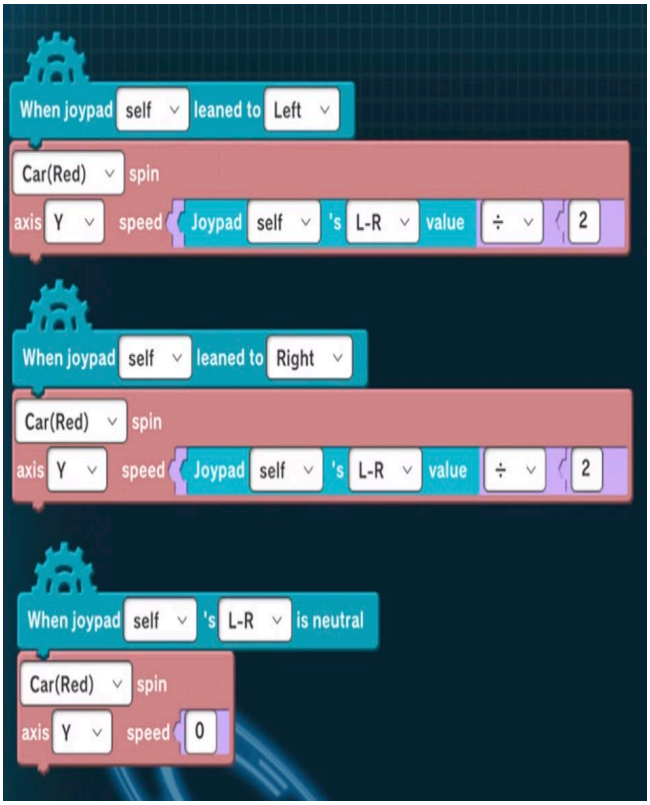



When you push the joypad L upward, the car will be moved in the z direction (direction of travel) by the up/down value of the joypad (maximum value: 1, minimum value: -1).

When the joypad L is operated upward, the car moves in the direction of travel. Move the car backwards with the joypad L controller downwards value.

When the joypad L is in the center position, the car moves in the z direction (moving direction) at 0, in other word, the force to move the car becomes 0. The car will move forward as long as the momentum continues and won't stop immediately.

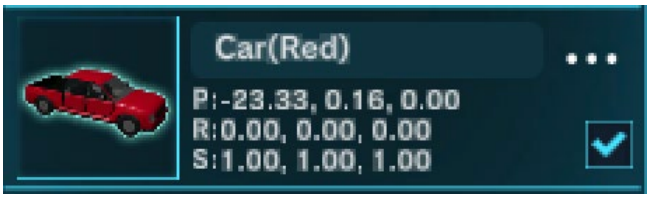
The figure below shows the program for Joypad R.

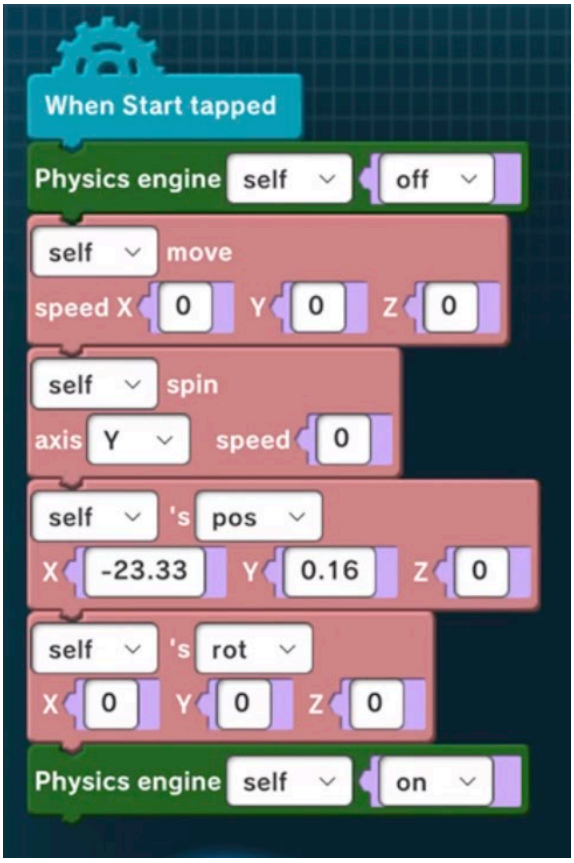



In this program, the joypad R is used to change the direction of the car. When joypad R is operated to the right, the front wheels of the car will turn to the right in the y direction by the value of the joypad (maximum value: 1, minimum value: -1). Negative values steer the car's front wheels to the left.

The left and right values of joypad R range from left (-1.0), right (1.0), center (0) using values from -1.0 to 1.0 to turn the wheels of the car. In this case, -1.0 to 1.0 is too large to respond, so as shown in this command (below figure), the value of the right joypad controller is divided by 2 and use the range that becomes -0.5 to 0.5.

Next, let's look at the program for the car object.





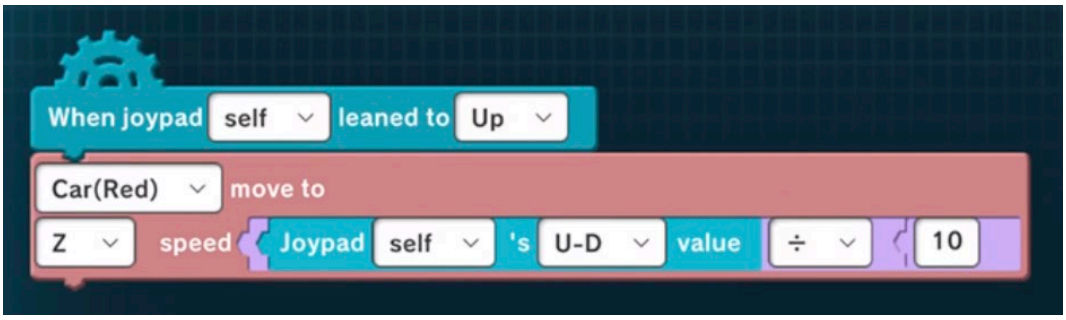

When the START button is tapped, the car will be placed in position on the circuit track.

Turn off the physics engine when moving the car to a given position. In the program on the left, the car is placed at the specified coordinates and the direction of the car is set by "self’s rotation". In the left figure, the car faces to upper direction of the screen aligned with the course at X:0, Y:0, and Z:0. You can also turn the car to the right of the screen by changing the value of the y coordinate.

Once the position and angle are set, turn on the physics engine.

## Advanced Challenge

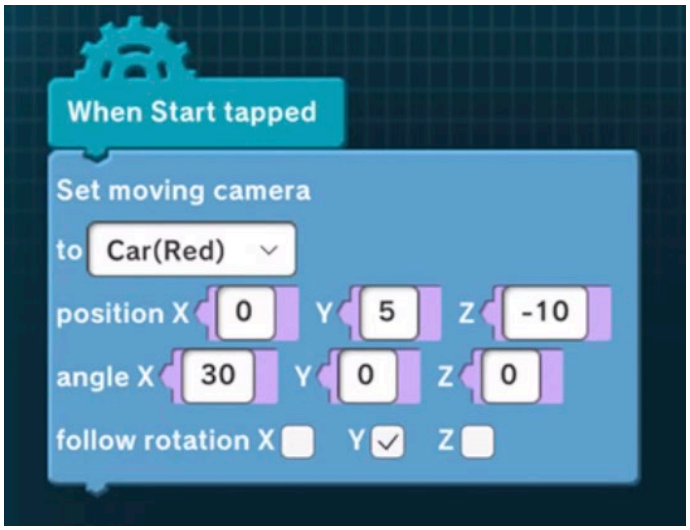
The joypad moves up, down, left, and right in the range of -1.0 to 1.0, but by changing the value, you can change the speed and operability of the car. The figure below shows an example of dividing the up/down value of joypad L by 10 to slow down the speed, which makes the driving easier. Try different values for the up/down and left/right of joypads to find a setting that is easy for you to use.



## Record and playback

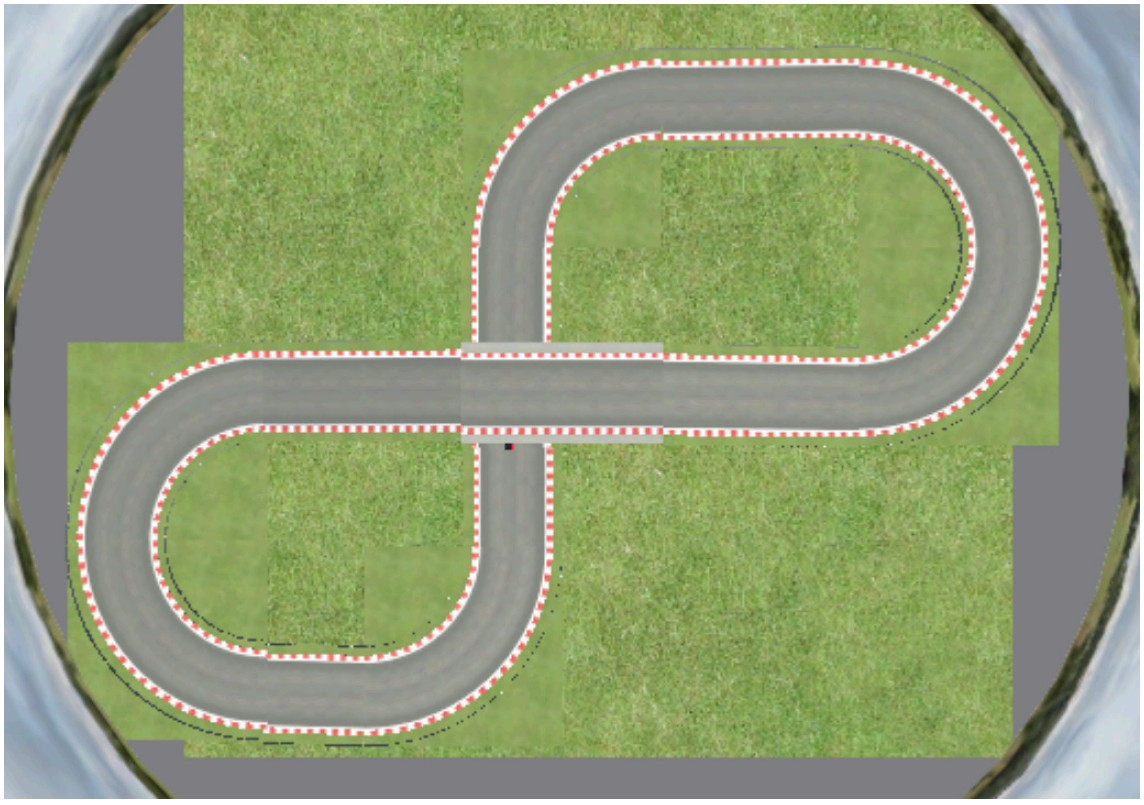
In Lab “Record and playback”, you can place the camera viewpoint behind the car to follow the car while enjoying the circuit driving. The figure below shows the lab opened and the start button tapped on the full screen.


Just like in the Lab "Car Operation", you will use the two joypads at the bottom of the screen to control the car and drive it around the circuit.

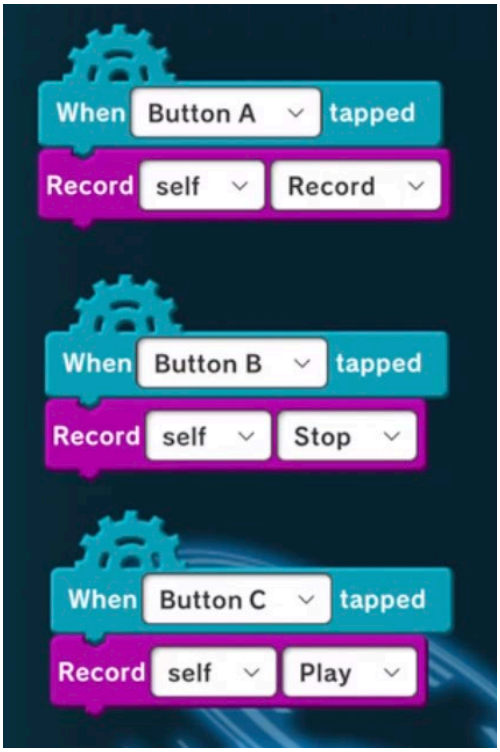



The viewpoint is changed by the program on the left in the camera object. To view the car from behind at a higher position as shown in the above figure, set the camera coordinates and angle as shown in the left figure. Experiment with changing the y, z coordinates and x values of the angle to see how the viewpoint changes. You can also look from the front of the car or look up from below.

The below figure shows the entire circuit of Lab “Record and playback” from above. Use it as a reference when you drive. You can see a multi-level intersection in the center.



In this lab, the program below has been added to the car object. With 3 commands, you can automatically record and playback the driving of the car.



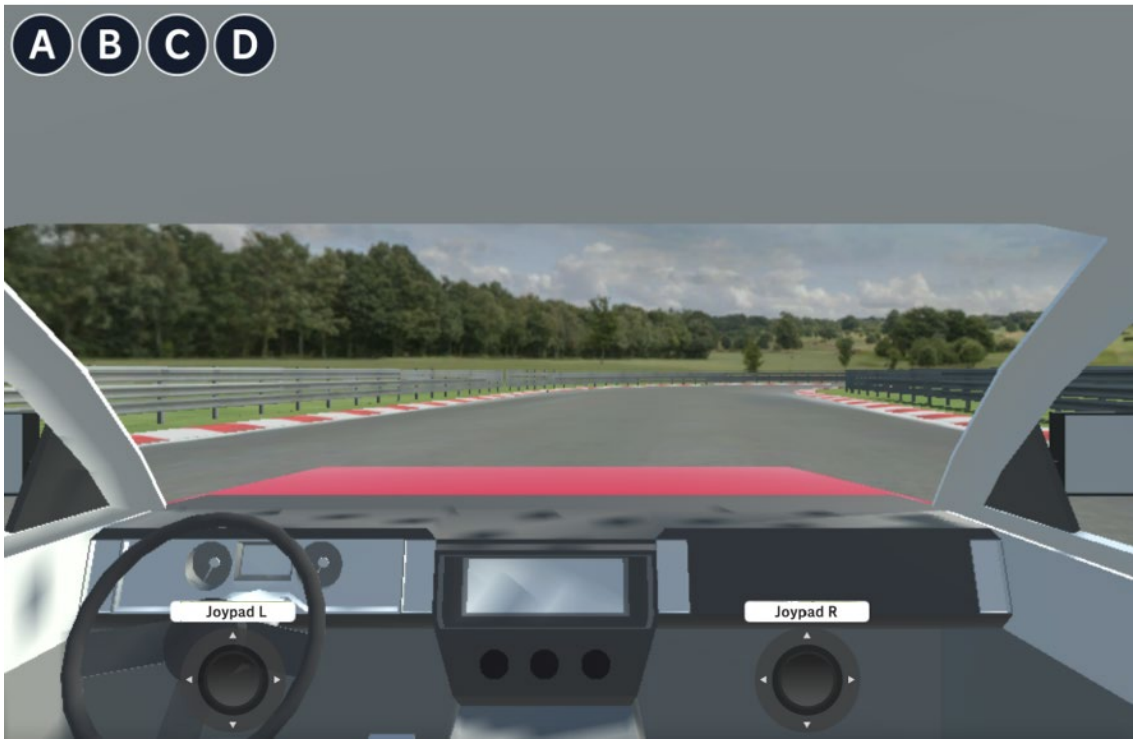

When button A is tapped, self (the car) starts to move and the recording is continued until button B is tapped. When button B is tapped, recording will end. Tapping on button C will start playing back what was recorded.

Recordable movement is only for movements that can be controlled with the joypad. The recording is not like video shooting, it is the operation data of the joypad that is recorded. Therefore, the playback may not exactly match the original motion.

The maximum recording time is 2 minutes.

## Advanced Challege

Let's change the viewpoint of the camera and drive while watching the view from the driver's seat. How can you set the camera to get the perspective shown in the figure below? Find out by changing the coordinates and angle of the camera.



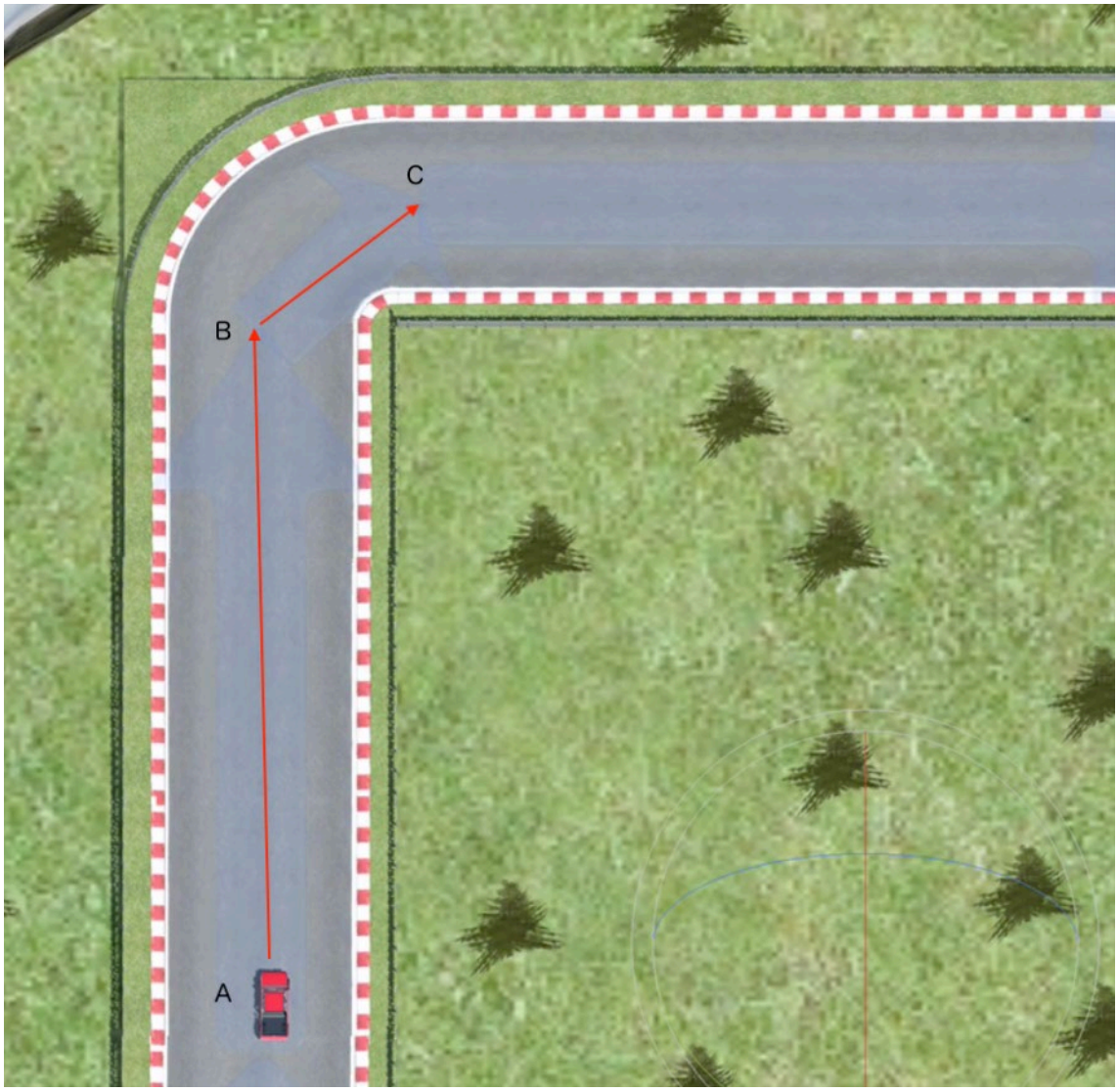

The figure below is a sample program. Please use it as a reference to determine coordinates and angles. In addition to the driver's seat, try various camera angles such as running with the camera on top of the car, placing the camera on the hood of the car, or following the car with a camera in the sky.

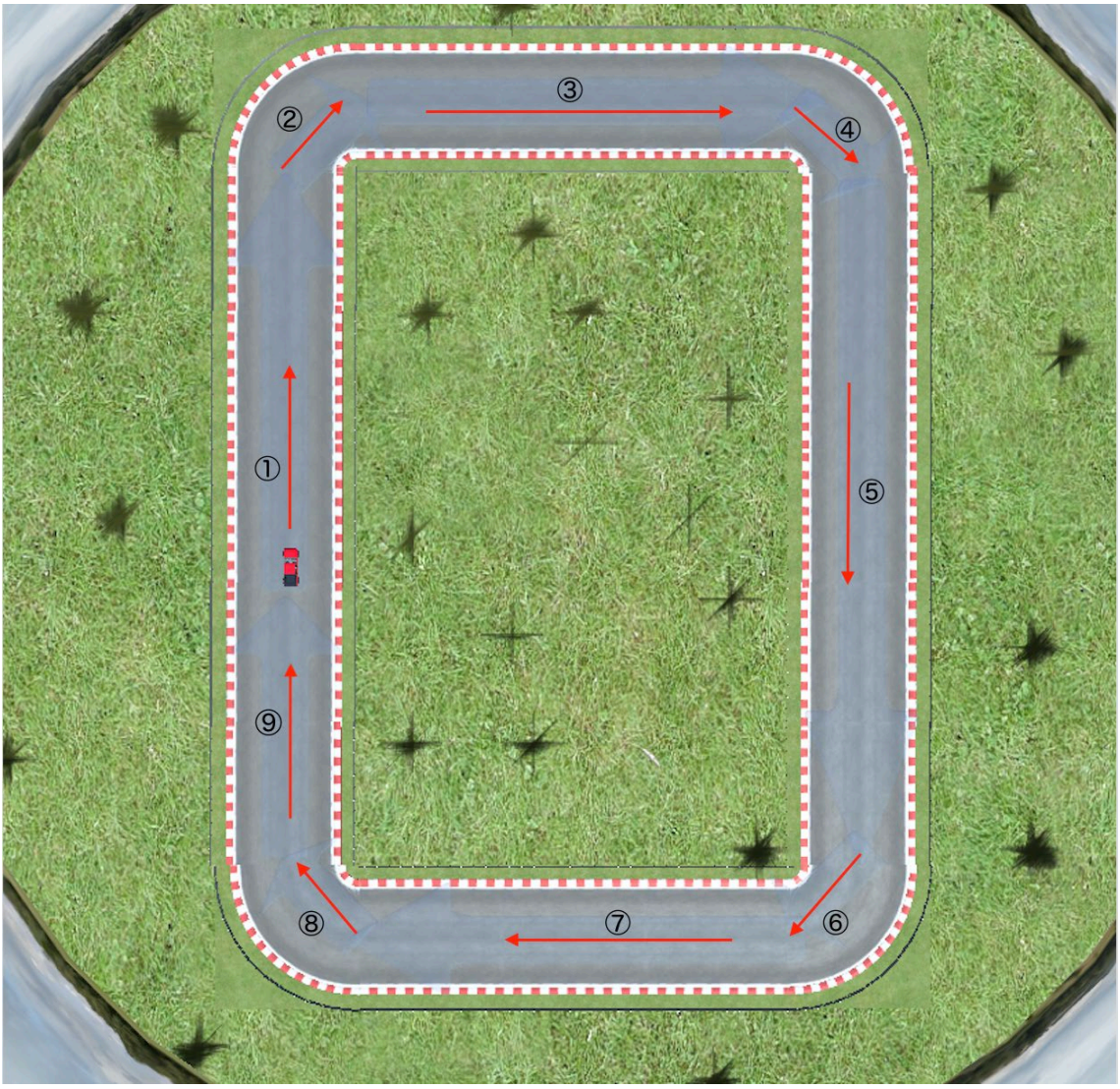
## Self-driving

In Lab “Self-driving”, the car will be automatically controlled to run the circuit. It is a rectangular course as shown below.

There are arrows placed along the course for automatic driving. The arrows are wide and light colored. By driving in the direction of the arrows, the car will run without deviating from the course. The figure below shows a part of the course.

The car moves from A to B (tip of the arrow) and from B to C (tip of the arrow). The red arrows are for illustration.



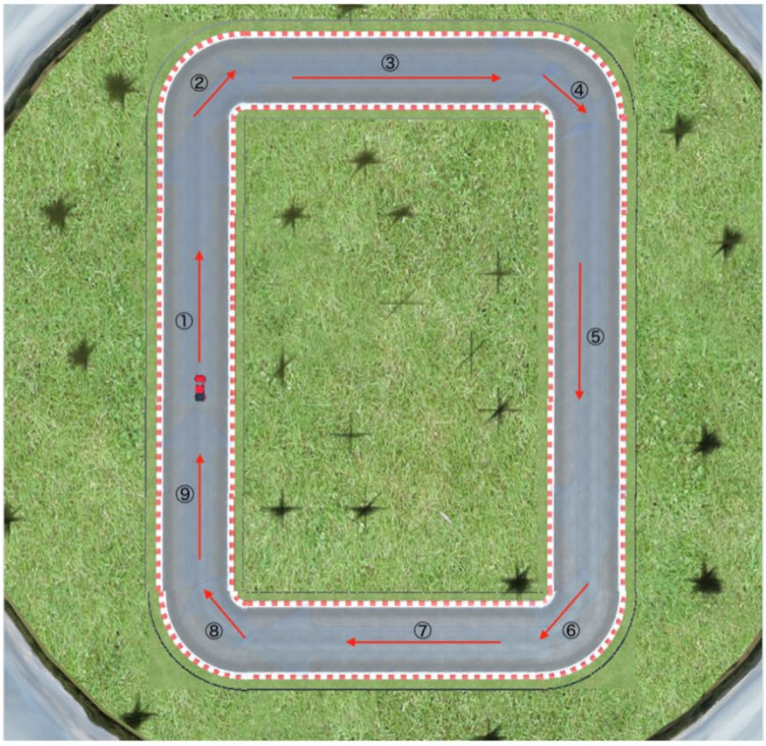



The arrows placed along the course are numbered from ① to ⑨.

In this lab, we use trigonometric functions to control the car's direction. When you use trigonometric functions, you need angles specific for trigonometric functions. The difference between the direction of objects such as "Arrow" and "Car” can be summarized as follows.

3D angle (arrow or car angle): angle measured clockwise starting from the north of the map

Trigonometric angle: angle starting at the east of the map and measured counterclockwise.




| | Trigonometric angle<br>90° - (A) | 3D angle<br>(A) |
|---|---|---|
| ① | 90° | 0° |
| ② | 45° | 45° |
| ③ | 0° | 90° |
| ④ | -45° | 135° |
| ⑤ | -90° | 180° |
| ⑥ | -135° | 225° |
| ⑦ | -180° | 270° |
| ⑧ | -225° | 315° |
| ⑨ | -270° | 360° (0°) |

In this lab, we use trigonometric functions.

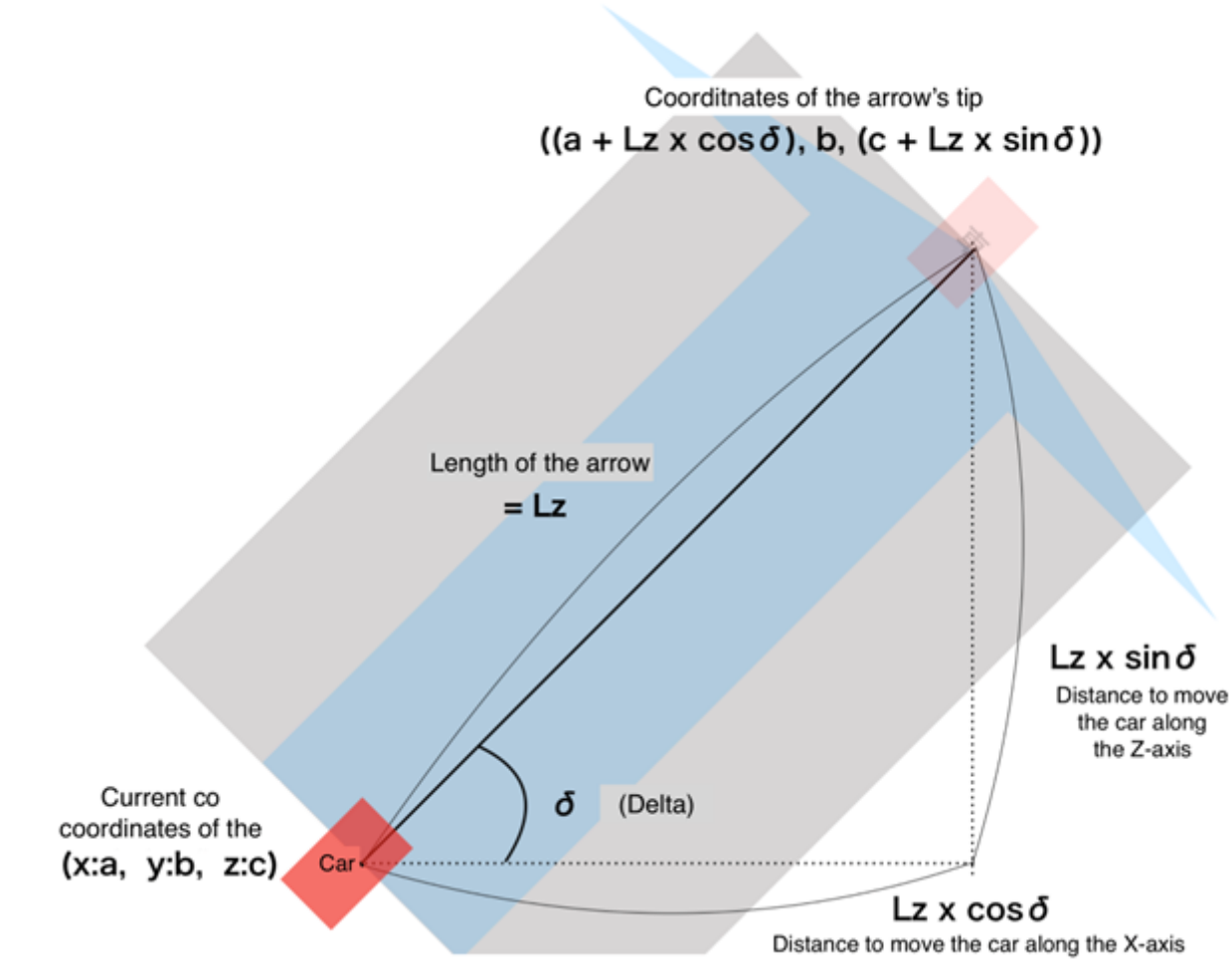
If the "trigonometric function angle" of the arrows ① to ⑨ in the table on the previous page is δ (delta), the method for realizing automatic driving is as follows.

When the car touches the arrow,

the direction of the car (the angle of the Y-axis) is set to δ (pointing in the same direction as the arrow).

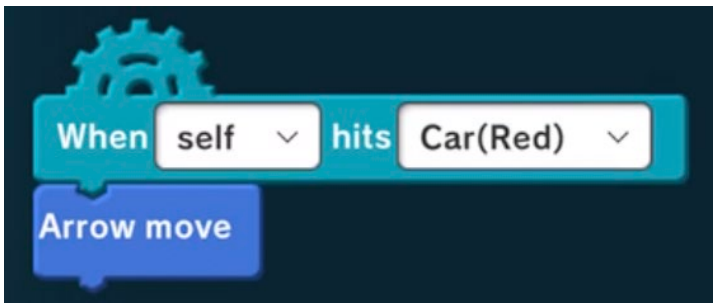
Move the car to the coordinates of the arrow's tip

The coordinates of the arrow's tip can be obtained using the trigonometric function as shown below.




In this lab, there is an "arrow" that will do the calculations shown above. The arrows are already laid out on the course, but you can also create your own program using these arrows. However, please note that it is valid only on the same horizontal plane (surface without height difference).

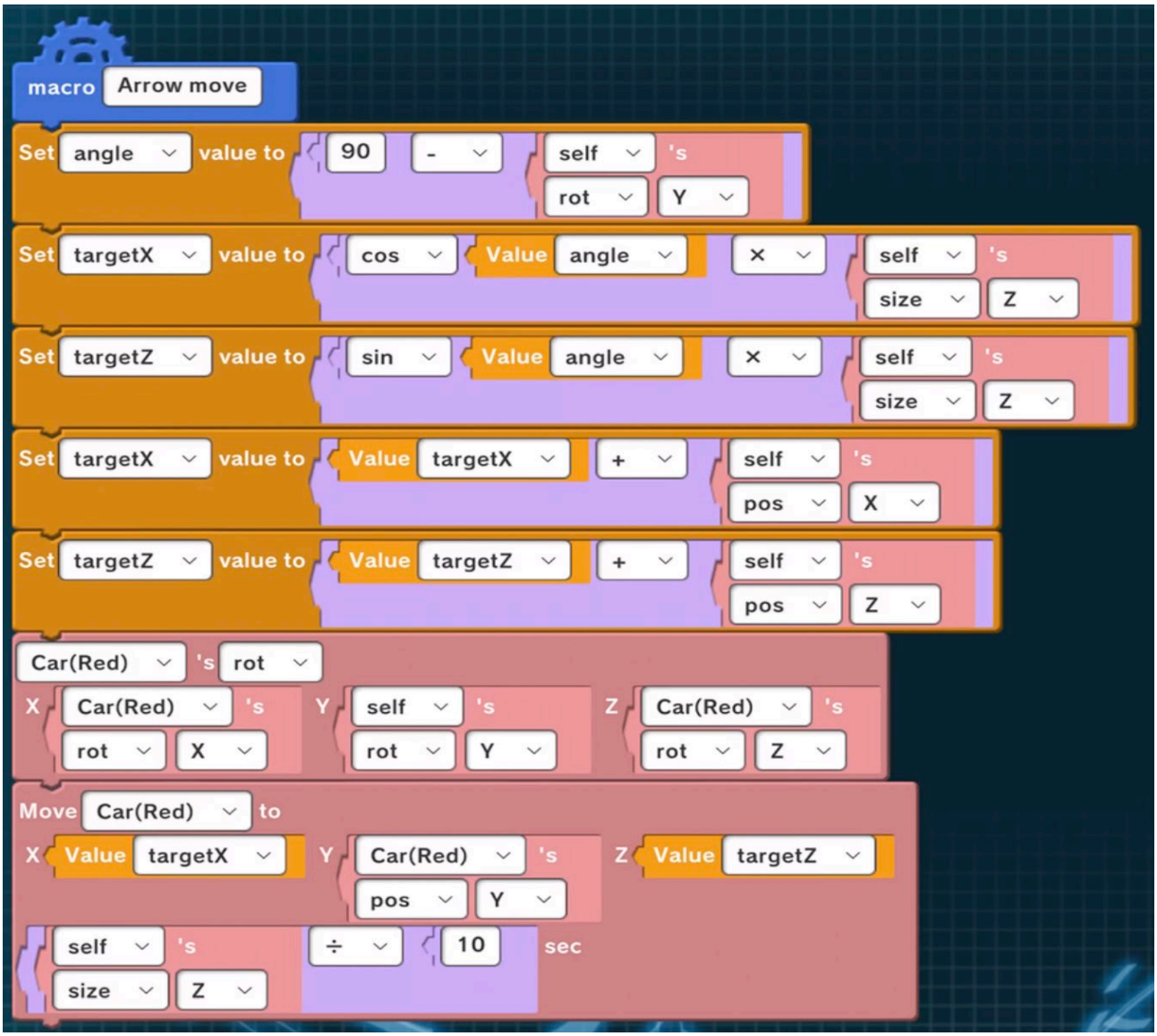
The program shown below is stored inside the nine arrow objects. When self (the arrow) come into contact with the car, "Arrow move" is summoned and executed. The "Arrow move" is a program for automatic driving, and is stored in an object called “Self-driving".



Let's take a look at the program "Arrow move", which is summoned by the arrow object. It is located in the ”Self-driving” object.


The entire "Arrow move" program is shown on the next page. This program uses a command called "macro". macro is used to create new commands using "Multiple commands". Determining what is actually processed in a macro is called "defining". A process once defined using macro can be used anywhere and anytime just by summoning the macro name, even in a large program. Using macros is very convenient because it saves you the trouble of repeatedly creating large programs.

The contents of the "Arrow move" macro command are defined as shown in the figure below. The content is a specifically implementation by program of the automatic driving method described above.




First command subtracts the "current Y-axis angle" from 90° to modify the "3D angle" of the arrow into a "trigonometric angle". There was an angle called δ in the figure on page 173, but the angle we are looking for here refers to δ.

angle (trigonometric angle) = 90° - 3D angle

The second command, the distance to move the car along the x-axis. The calculation formula is as follows.

targetX = cos(angle) x length of the arrow

The third instruction, the distance to move the car along the z-axis. The calculation formula is as follows.

targetZ = sin（angle） x length of the arrow

The fourth command adds the "current x-coordinate of the arrow" to the target X obtained earlier to calculate the actual x-coordinate that the car should aim at.

target X = target X value obtained earlier + self (arrow) x-coordinate

The fifth command adds the "current z-coordinate of the arrow" to the target Z obtained earlier to calculate the actual z-coordinate that the car should aim at.

target Z = target X value obtained earlier + self (arrow) z-coordinate

*The reason why only the x and z axis directions are calculated above is that the y axis direction is vertical (up and down). Here we are only considering horizontal plane motion.

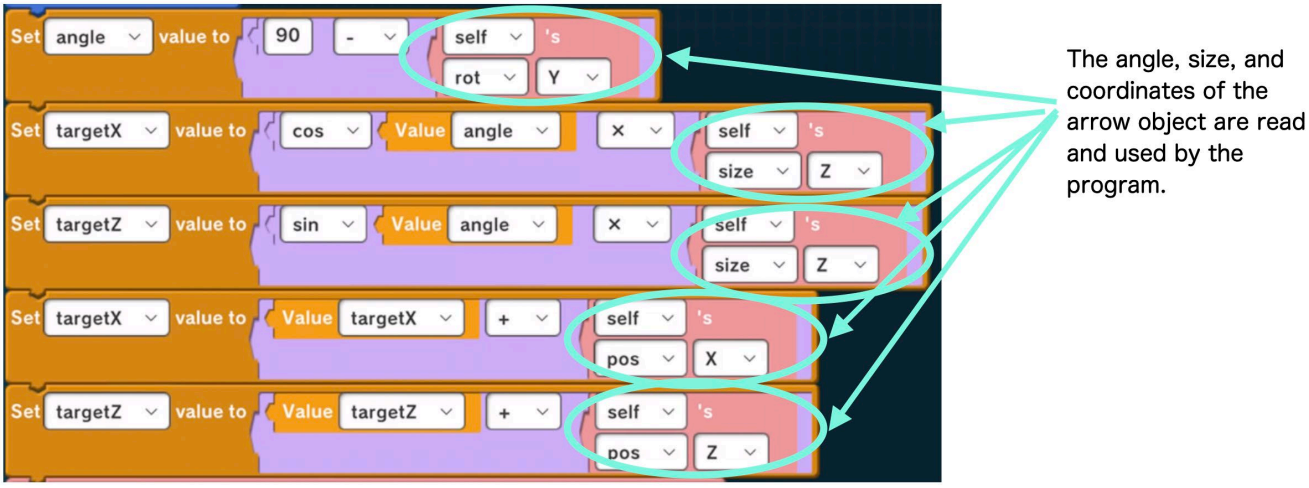
The sixth command is for aligning the "direction of the car" with the "direction of the arrow". It sets the angle of the car to self (arrow) angle.

The seventh and final command is to move the car to the x-coordinate (targetX) and z-coordinate (targetZ) obtained above. The time required for the move is specified by

Self (arrow)’s z-axis size ÷ 10 (seconds)

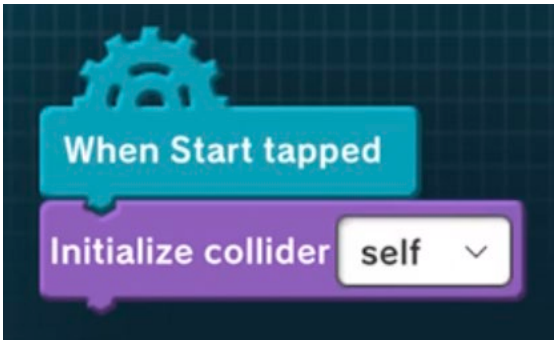
The division by 10 is to adjust the speed. The larger the dividing number, the faster, and the smaller the dividing number, the slower.

In addition, in this macro command, calls and uses the coordinates, size, and angle when the arrows are placed on the course, so you don't have to place a pin to check the coordinates or size every time you place nine arrows. The arrows can be placed anywhere on the course, in a straight line or a curve. Arrows can be relocated or changed in length without programming.




These tricks can save you time and effort.

The nine arrow objects include the program shown below as well as the program that summons the automatic "Arrow move" described above.




When the START button is tapped, the program first initializes its own collision detection. This is because the racecar was placed on top of the arrow at the start, so the racecar and the arrow were already in contact when the program started running. This command is necessary because in this automatic driving, the subsequent processing is determined based on whether the car is in contact with the arrow.

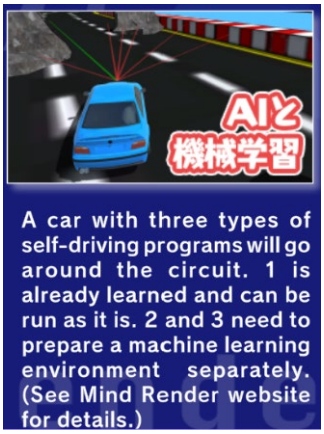
Once you understand how self-driving works, you can run it and see how the car drives automatically. You will see how the car drives towards the tip of the arrow on every arrow.

## Advanced Challenge

Challenge a program like the one below.

- Create a program that adds and angle to the x-coordinate of the course to correspond to the height difference of the course
- Change camera's viewpoint by tapping the A to D buttons
- - In Lab 6-5, you can create your own course. Put the arrow of Lab 6-3 on the course you made and try to challenge automatic driving

In addition, in the lab "AI and Machine Learning", you can see autonomous driving that implements an AI program that allows the car to avoid obstacles on its own. Please enjoy AI automatic driving while switching viewpoints by taping the A, B, and C buttons.




The AI automatic driving program is made using Mind Render's related product "Mind Render/AI Drill". For a detailed explanation of "Mind Render/AI Drill", please visit the following website.。

https://mindrender.jp/

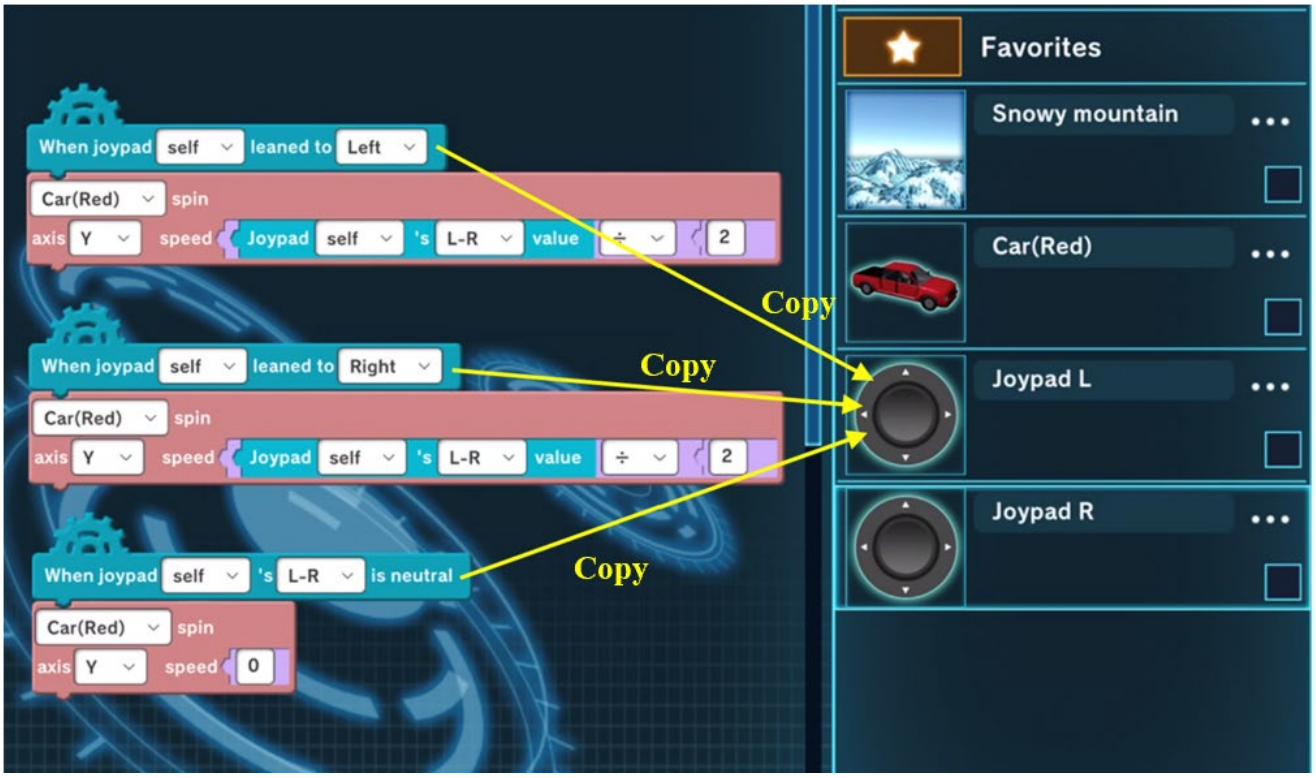
## Snowy Mountain

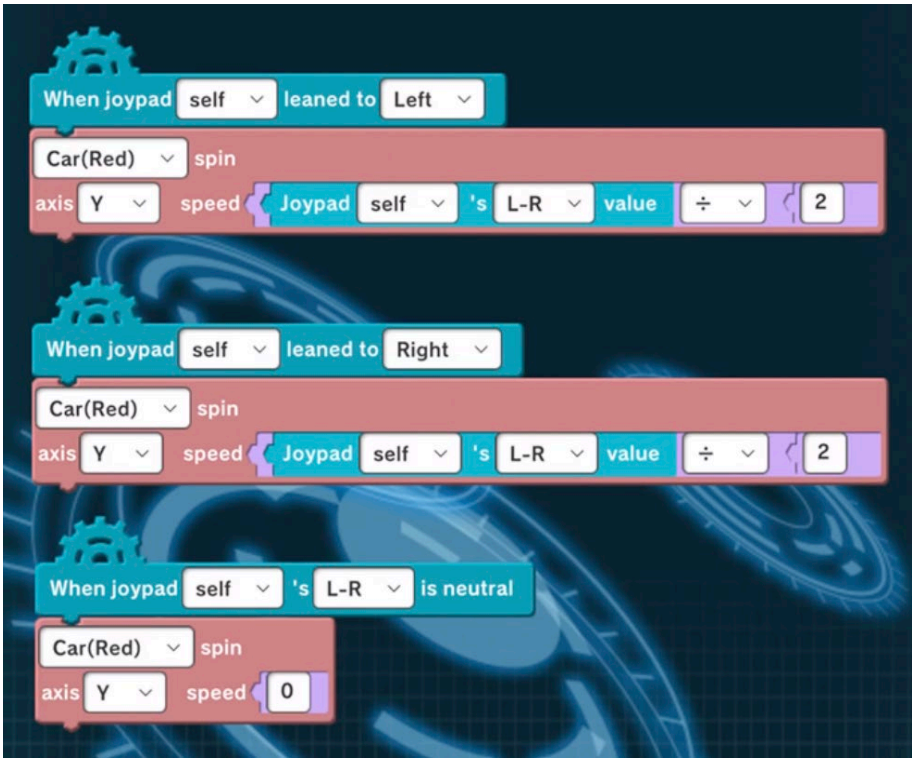
In Lab "Snowy Mountain", you will drive around the snowy mountains. Enjoy the view of the snow-covered mountains as you drive. When you open the lab in full screen mode, it looks like the image below.


Use the two joypads to control the car same as in the lab "Car Operation" and "Record and Playback".

In order to make this lab easier to use on a PC, let's change the program so that it can be controlled with just one joypad. Joypad L uses only up and down movements, and Joypad R only uses left and right movements, so you can combine up, down, left, and right movements with a single joypad. This section explains how to combine Joypad R programs with Joypad L.

Figure below shows when JoypadR object is opened. Copy the three programs to JoypadL by drag-and-drop.





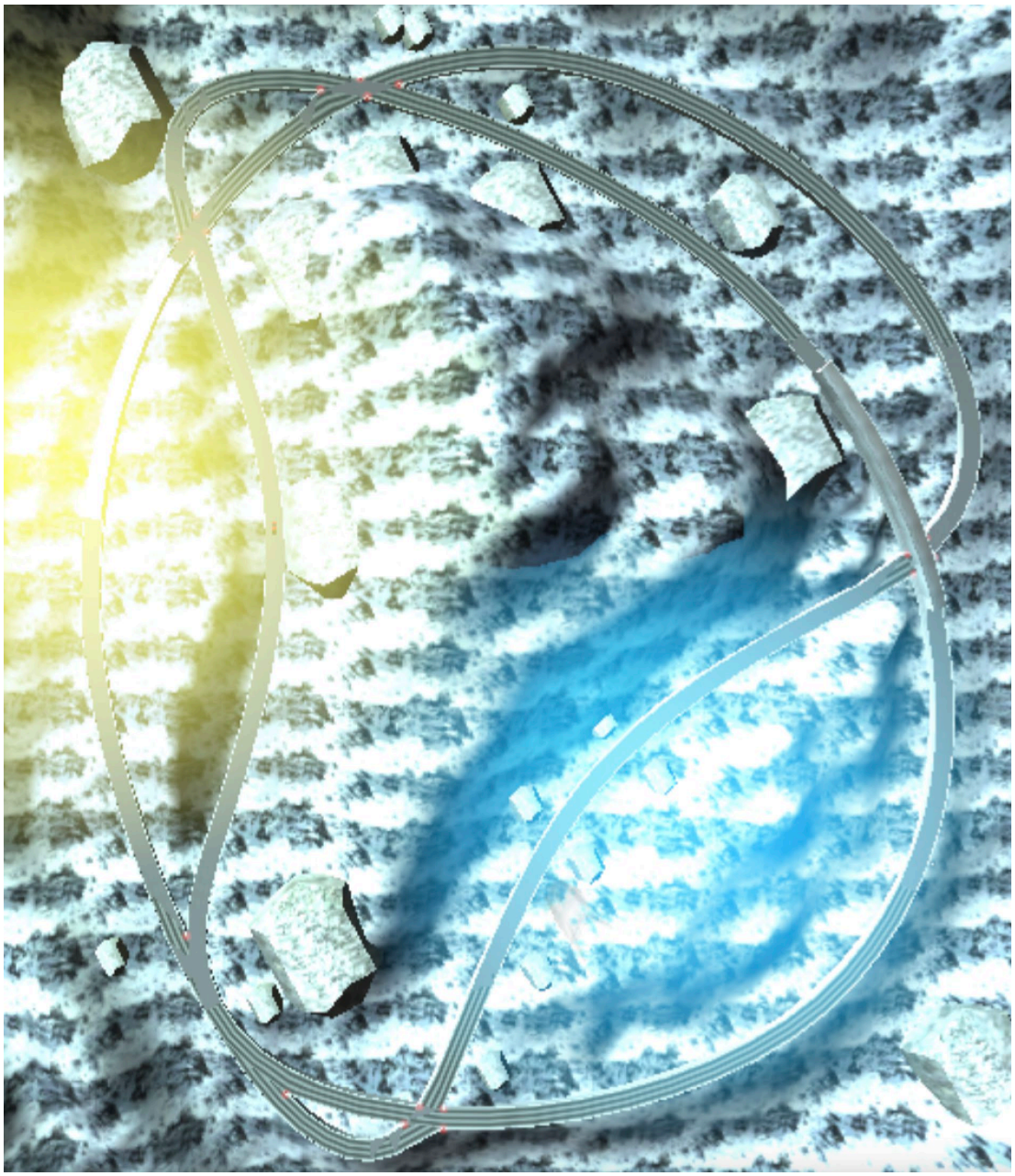

After copying, check if there are three programs in Joypad L as shown on the left.

You can now move forward or backward by moving Joypad L up and down, and turn left and right by moving Joypad L left and right.

## Advanced Challenge

Once you've gotten used to driving on the snowy mountain course, try placing various obstacles on the course to enjoy a thrilling ride. As an example, in the figure below, an abandoned car is placed just after the start position.

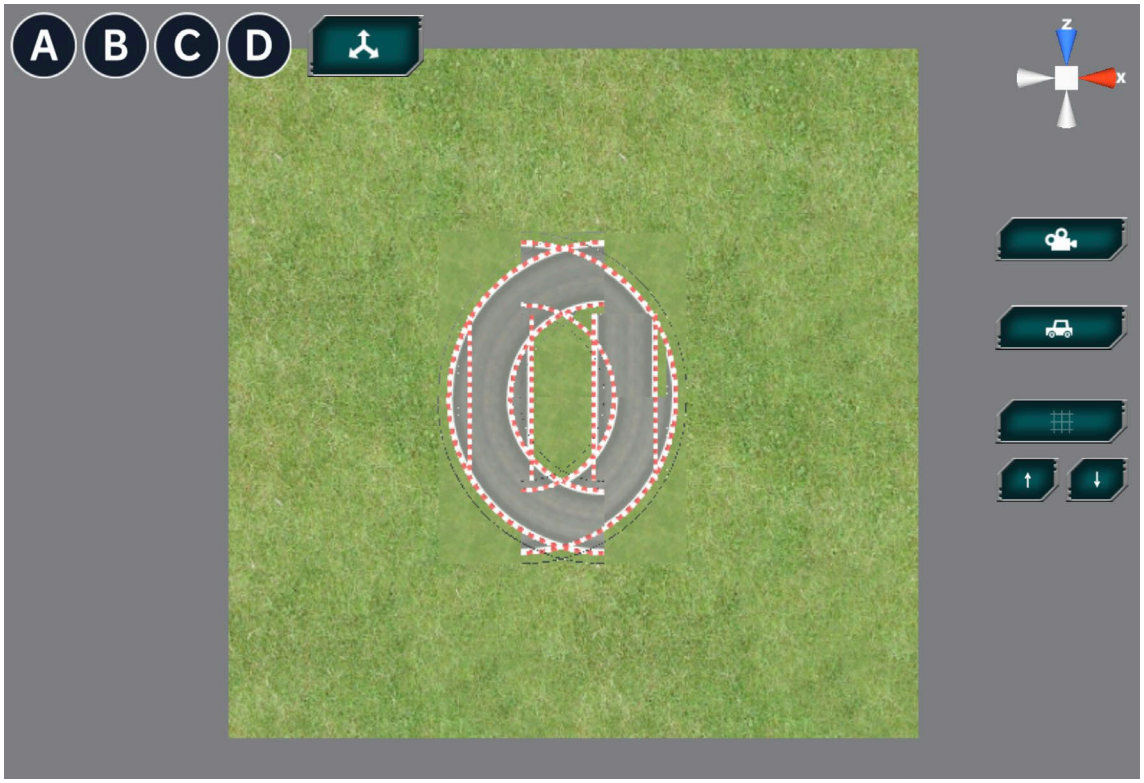
The figure below shows the entire snowy mountain course. Look at the whole course and place various obstacles to make the driving more exciting.



# Course making

In Lab “Course making”, you can build your own circuit course.

In this lab, course parts for creating your own circuit course are pre-registered. If there are not enough course parts, you can add as many as you want from the Object menu.
The figure below shows when the lab is opened in full screen mode.

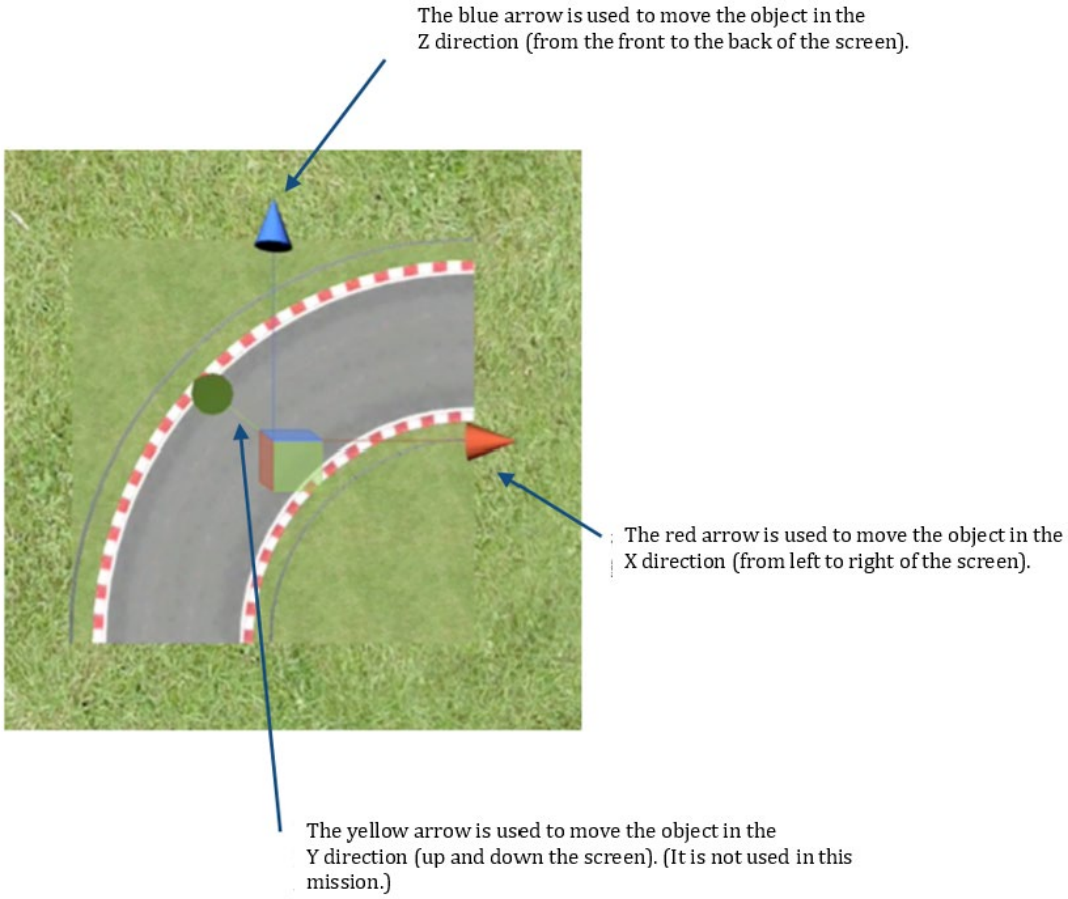



Here, let's build a circular circuit course as a simple example.

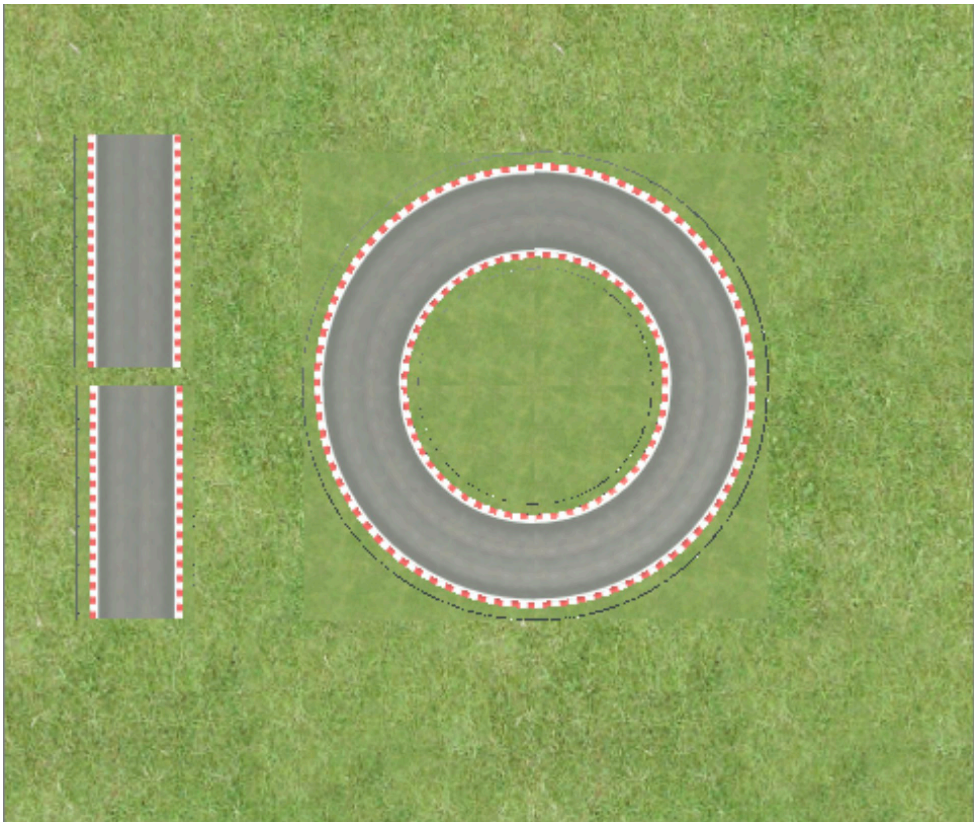
All the course parts can be moved on the lawn。The procedure for arranging course parts to create a circuit is explained below.

1）Make sure that the buttons next to the A, B, C, and D buttons that appear when the program is stopped are icons that indicate "Move" as shown below. If it shows another icon, tap the button until the "move" icon is displayed.

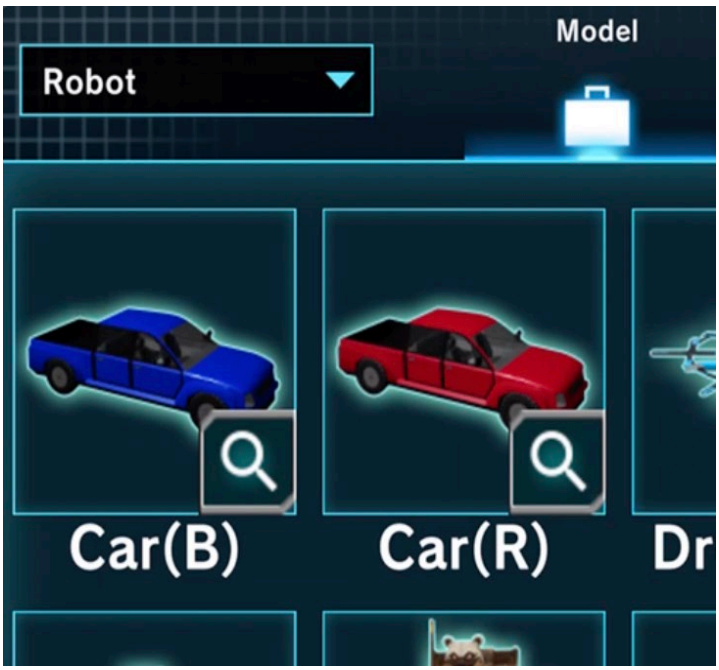
2） When the button shows the "Move" icon  , tap the course component you want to move, and a 3D arrow will appear as shown below. This arrow is to move objects.



3）A circular course can be created by combining the 4 curved course parts provided in this mission using the 3D arrows.



4）Add your favorite car from the object menu.



5）Let's make the car object’s program by yourself referring to the previous missions。

For example, if you add joypad controllers and write programs as shown in the figure below, you can move the joypad as you did in the Lab "Car operation" "Record and playback" "Snowy Mountain" and control the car.

In addition, you can also challenge automatic driving by placing the arrows used in the lab "Self-driving".


## Advanced Challenge

Let's race with a computer-driven self-driving car. You can use the self-driving arrows used in the lab "Self-driving", but let's use a self-driving car that use AI here. If you select "Other" in the Model tab of the Object menu, you will find an object called "AI Car". This object is pre-loaded with a program for automatic operation using sensors. Place this AI car on the course and run it and put some obstacles on the course such as soccer balls. The figure on the next page shows an example of the completed and executed program. Use one joypad controller to control your car.


A
B
C
D
Controller


Controller

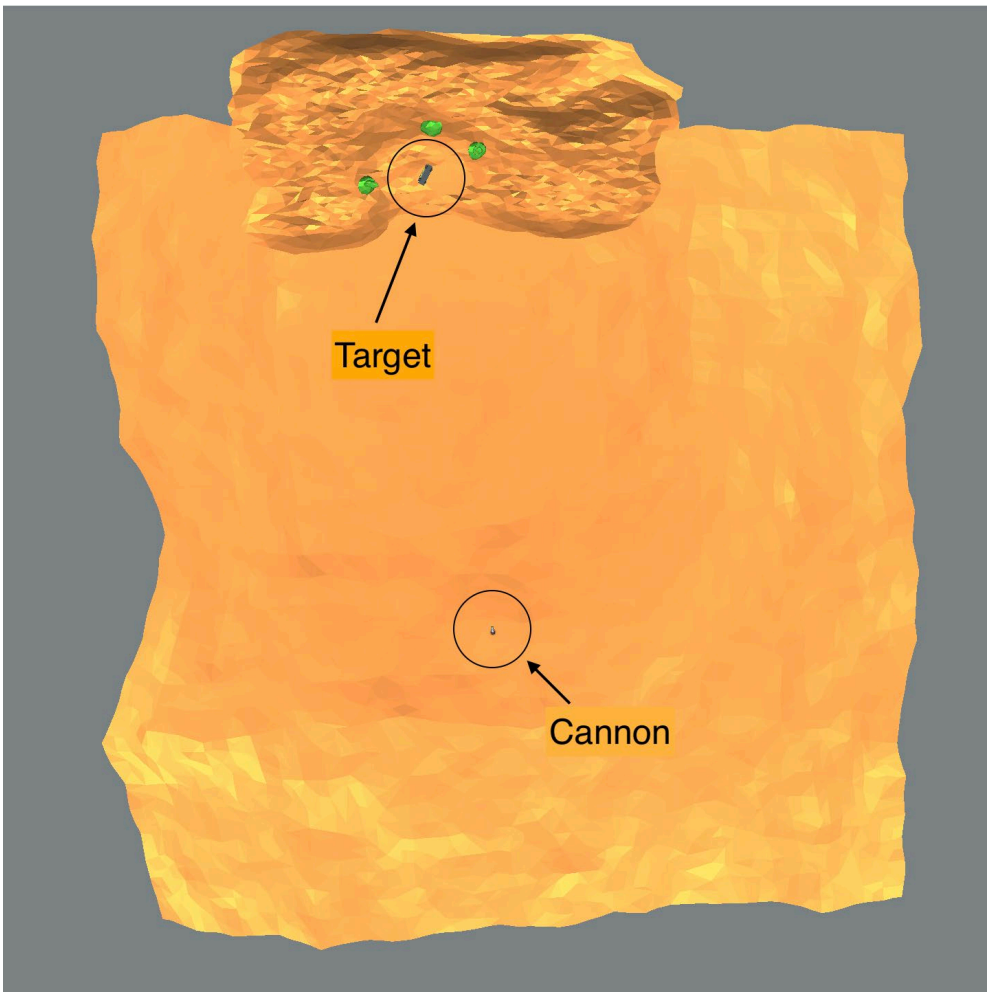
# 11. Let’s study a little 4: “Shoot a Cannon“

## Shoot a Cannon

In Lab “Shoot a Cannon“, shoot a cannon and aim at a target. Can you successfully hit the target in various conditions such as blowing wind and wall obstacles?

# Shoot a Cannonball 1

In Lab "Shoot a Cannonball 1", you shoot a cannon at a target. The target is a scrap car. Check the approximate distance from the cannon to the target in the figure below.

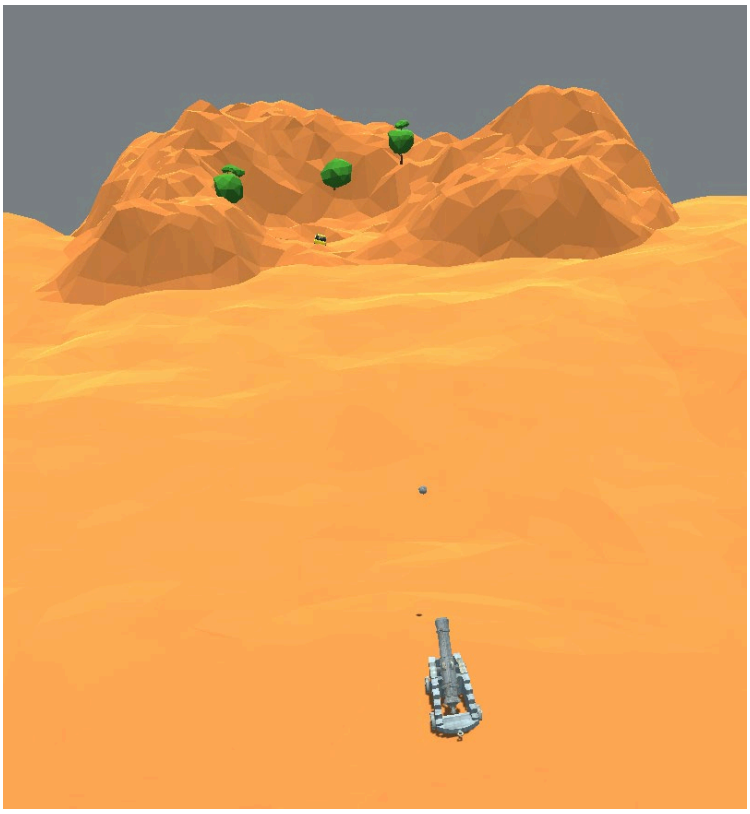


In this lab, you will use the A and B buttons to control the cannon.

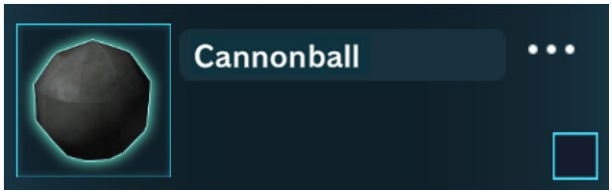
A button: place a cannonball in the cannon (place the cannonball in the cannon barrel)

B button: shoot a cannonball


The figure below shows a cannonball being fired by taping the B button.

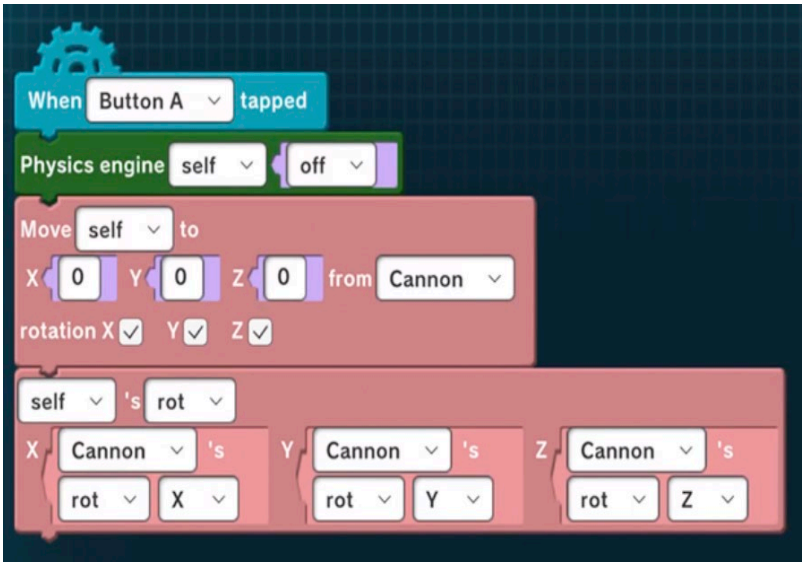


When you tap the B button to shoot a cannonball, the cannonball will not reach the target.

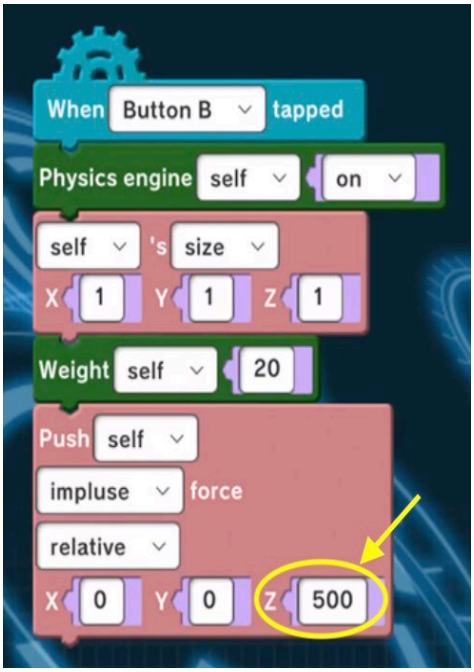



Let's modify the program to hit the target.

The program that shoots out the cannonball is stored in the Cannonball object.




The left figure is a program that puts the cannonball into the barrel when A button is tapped. The barrel looks like the real thing, but you can't really keep a cannonball inside. Placing a cannonball in the cannon is like keeping the cannonball in the air. Turn off the physics engine so that the cannonball doesn't fall to the ground. The bullet's direction aligns all x, y, and z with the barrel's direction.



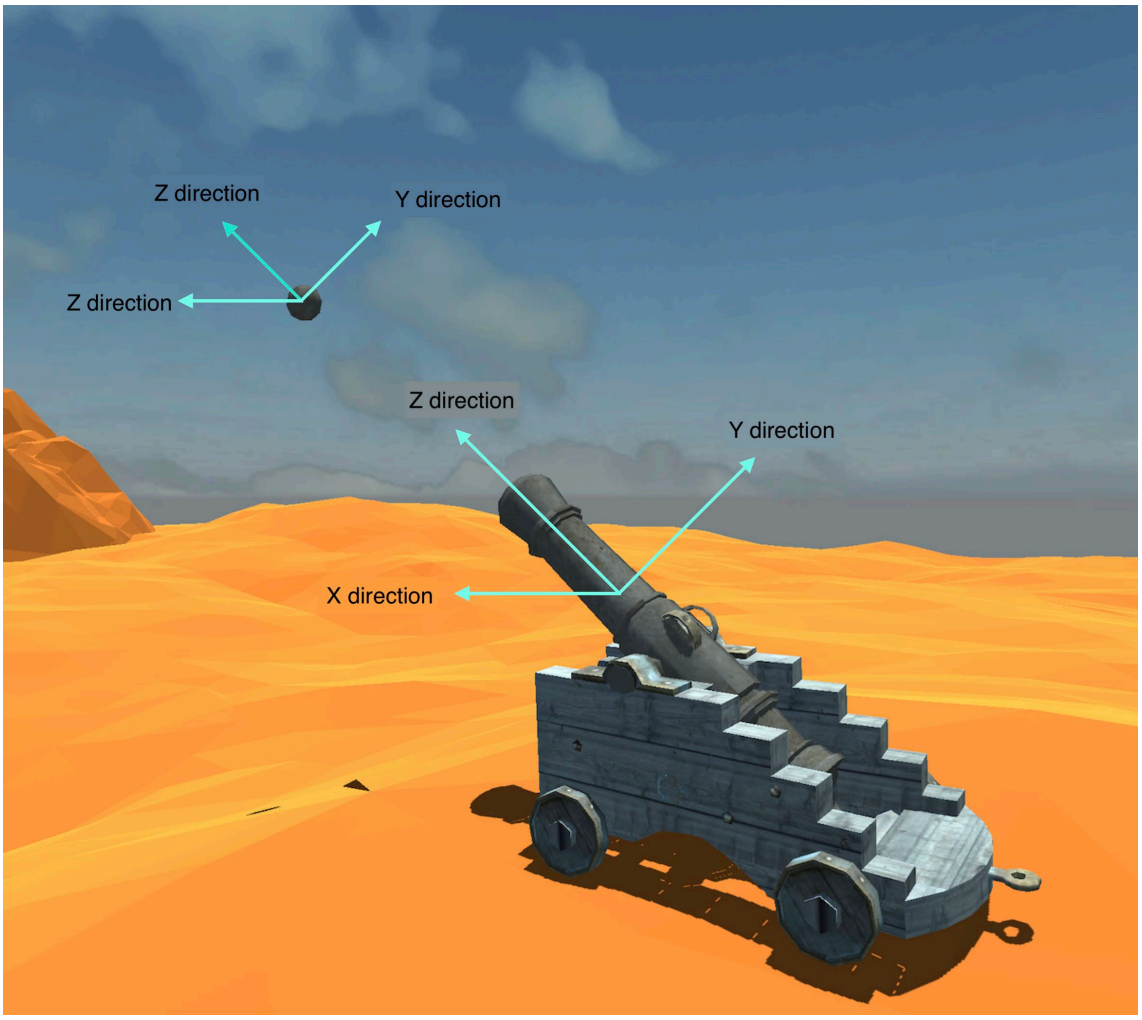

The left figure shows a program that shoots a cannonball by tapping the B button.

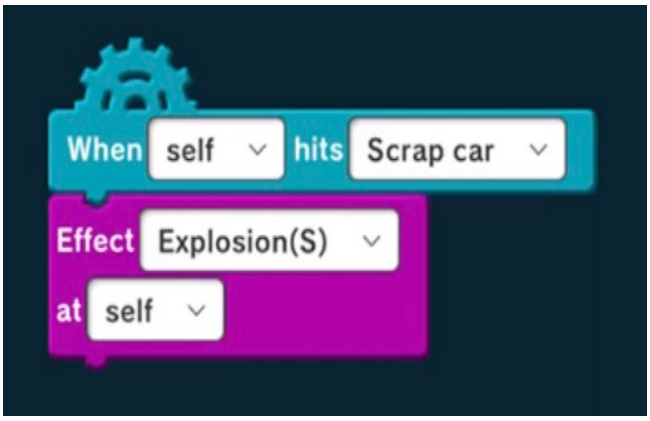
A real cannon would use the explosive power of gunpowder, but here we add "impulse force" to the cannonball and shoot it out. Turn on the physics engine to enable the impulse force. Set the size and the weight of the cannonball, and how much impulse force will be applied in the Z direction (direction of cannon barrel).

To change how far the cannonball flies, change the numerical value of the impulse force (Z direction) indicated by the arrow in the left figure. If you increase the number, the cannonball will fly farther because it will be pushed with a greater force.

The "relative" setting means that the force is applied to the direction of the cannonball itself. If you switch "relative" to "absolute", the direction of the force is determined based on the direction of absolute coordinates regardless of the direction of the cannonball itself. Note that button B can be pressed even if there is no cannonball in the barrel, this means that the A button must be pressed to load the cannonball in the barrel after each shot to fire the cannonball in the correct position.

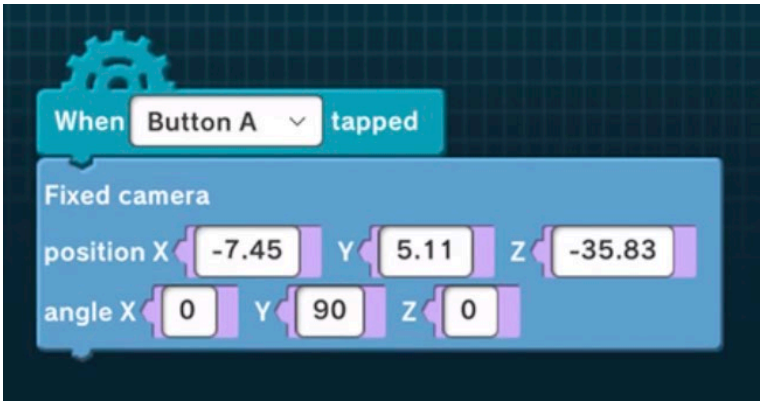
Please refer to the figure below for how to specify the direction of the cannonball using the x, y, and z coordinates.。






There is another program for cannonball as shown on the left. This program is used to play the explosion effect when the cannonball hits the target scrap car. In the program on the left, the effect is specified as "Explosion(S)", but you can also add and specify other effects from the "Effect" page of the object menu.

In Lab "Shoot a Cannonball 1", the viewpoint of the camera changes in the middle of the program. Let's examine the camera program.




When the A button is tapped, the camera's coordinates are set as follows:

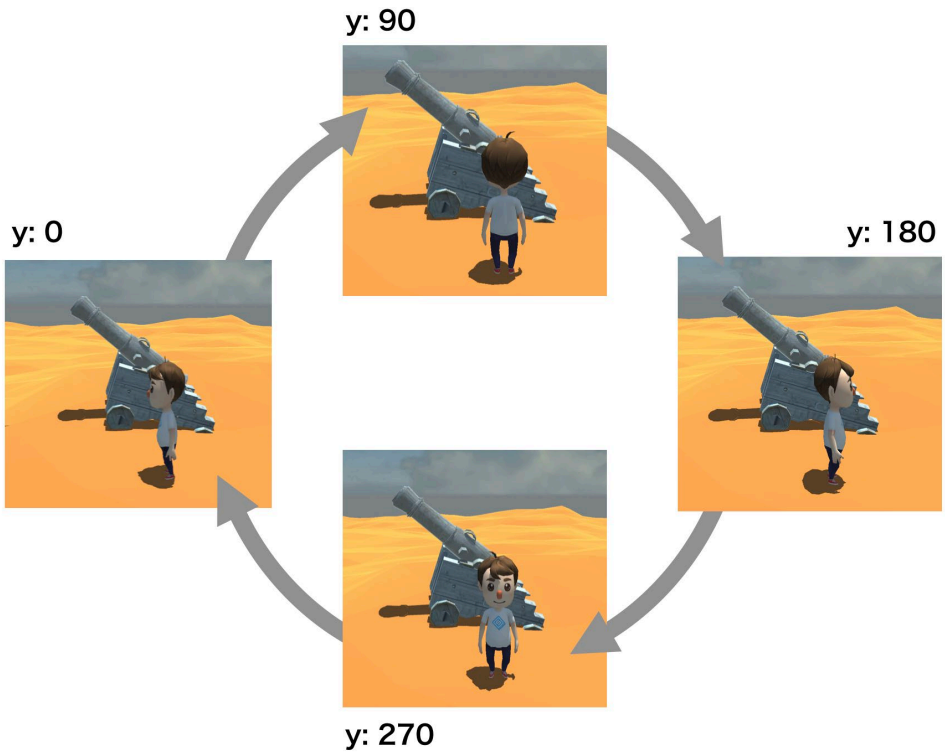
x: -7.45

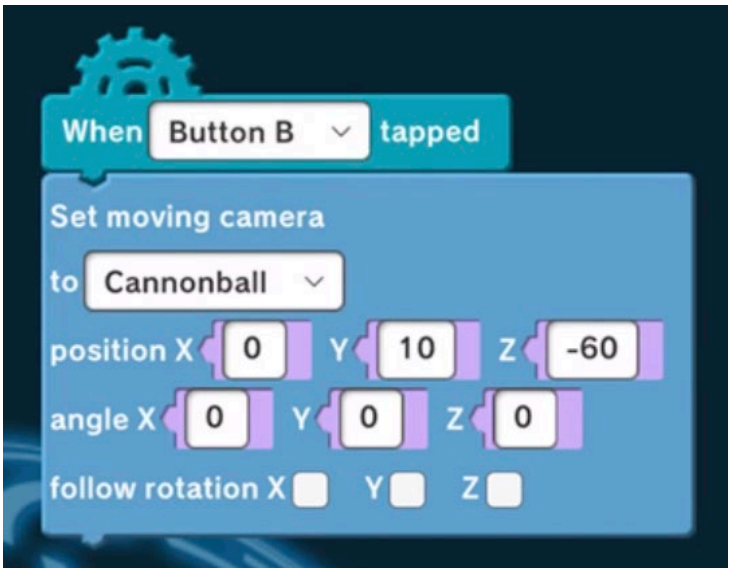
y: 5.11

z: -35.83

A fixed camera is a camera that is fixed at a certain position. The angle setting y: 90 is to rotate the camera around the y-axis (the axis extending from the ground to the sky) and look at the cannon.

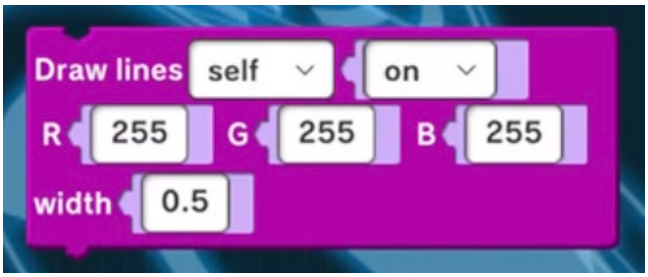
As for the relationship between the positions of the camera and the cannon, imagine that the camera is at the boy's position in the figure below. You can see that the camera looks towards the cannon when the camera angle is y:90.




When B button is tapped, the camera settings will change from fixed to moving, and the mounting position will be specified as “Set moving camera to Cannonball”. Coordinates and angles specify how far and which direction the camera should be mounted to the cannonball. In the left figure, Y:10 is slightly above the cannonball, and Z:-60 is behind the cannonball. You can move the camera closer to the camera or change the mounting position to look up at the cannonball from below. If you check "follow rotation X, Y, Z", you can also set the camera to rotate when the cannonball rotates.

## Advanced Challenge

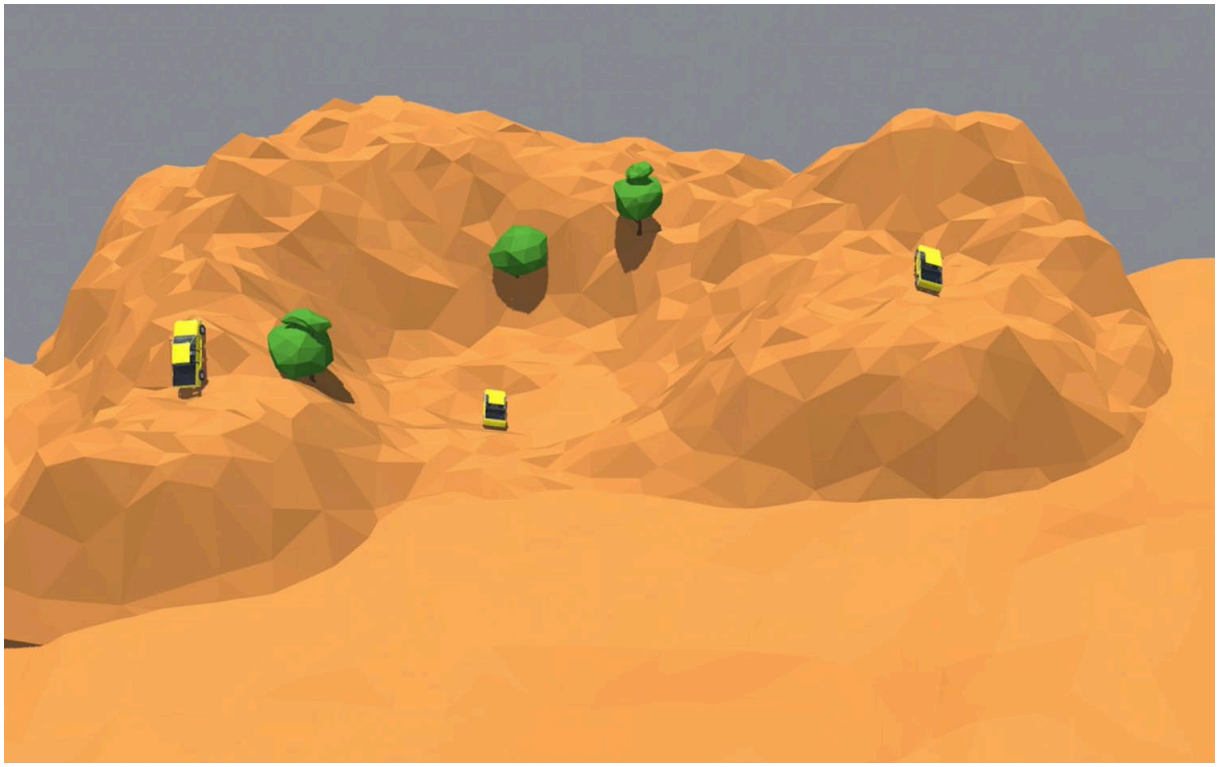



The trajectory changes when you change the value of the impulse force. To see how the trajectory changes try repositioning the camera and looking for coordinates and angles where you can see the trajectory from beginning to end. To further visualize the trajectory, you can draw a line to trace the path of the trajectory. Use "Draw line” command in the "Effect" category to do this.

# Three Targets

In Lab "Three Targets", there are three target cars. Let's make a program that aims at three targets and hits them one after another.

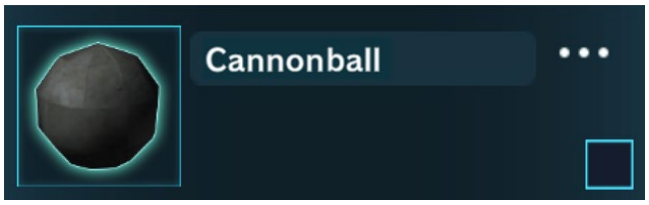
On the hill, there is one target scrap car on the left, one in the middle, and one on the right. The figure below shows the location of the three scrap cars.



The program is like the followings.

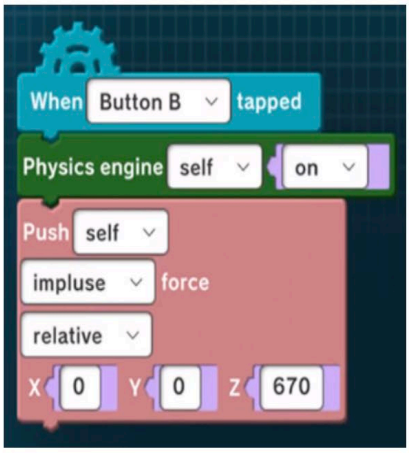
- Tap A button to set the cannonball in the cannon (same as Lab “Shoot a Cannonball 1”).
- Tap B button to fire a cannonball at the center target.
- Tap C button to fire a cannonball at the left target.
- Tap D button to fire a cannonball at the right target.

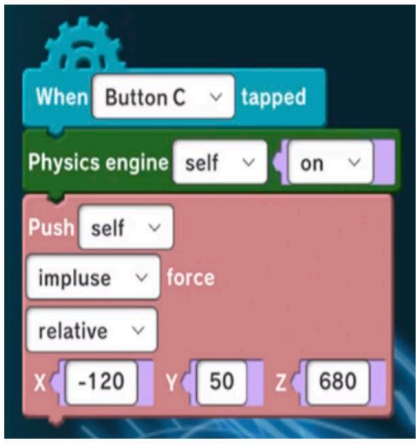
Press the A button each time you fire to set the cannonball and then aim at the next target.

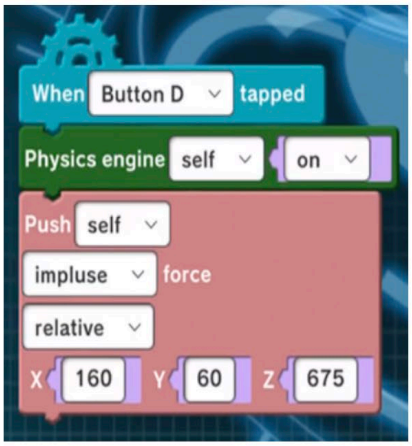



Let's see how the settings look like in the Cannonball object’s program.

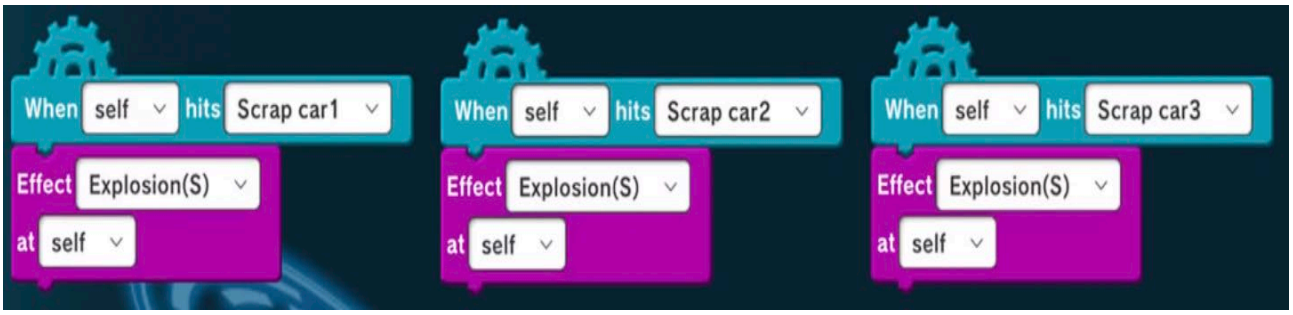
The settings for shooting the three targets are shown in the figure below. Please note THE X, Y, and Z settings for the impulse force on buttons B, C, and D, respectively. These settings are just an example. It is possible to hit the target with other settings, so let’s try it.



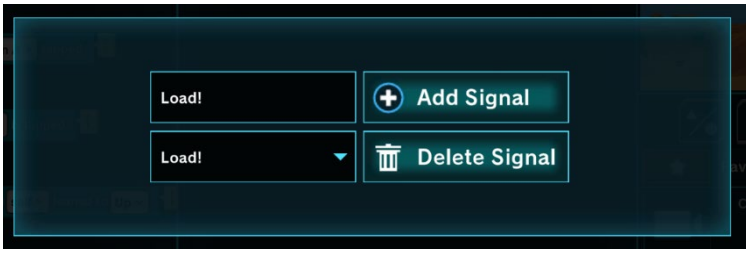
The effect when hitting the target also needs to be set for each target. Please refer to the figure below.

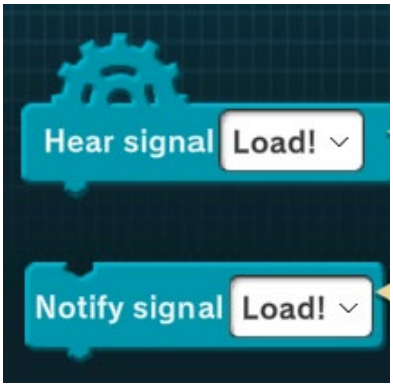



## Advanced Challenge

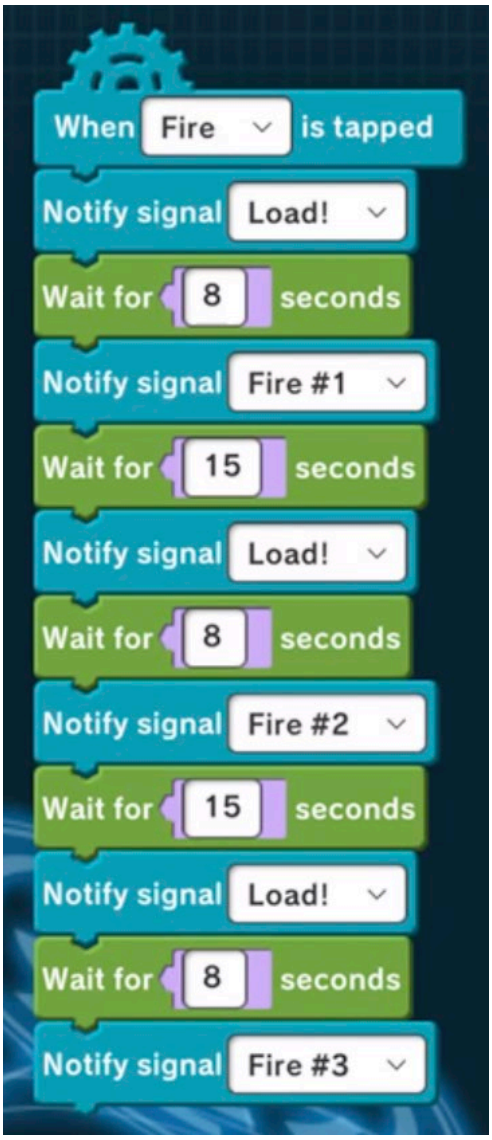
Let's try a program that automatically aims and shoots three targets one after another. As an example, here's how to do it using Signal. Try it for real yourself. Signals are cues, and some signals are decided in advance in Mind Render and used by sending and receiving signals as necessary. In this advanced challenge, to load the cannonball into the cannon (to place them in the designated position), send the signal "Load" or send another signal "Fire #1" to fire the cannonball at the first target, and so on.


To set up a new signal, click "Make Signal" at the bottom of the block category (left). Next, enter the name of the new signal you want to set in the signal setting screen (below) and click the "Add signal" button.

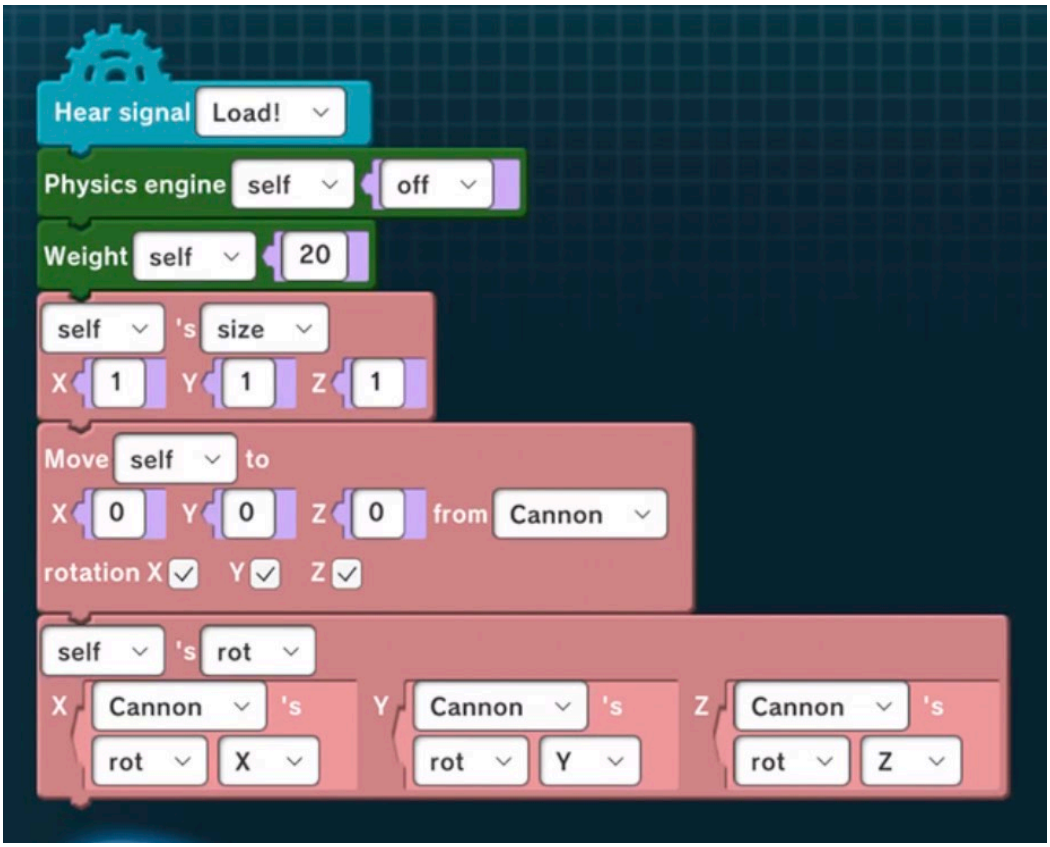




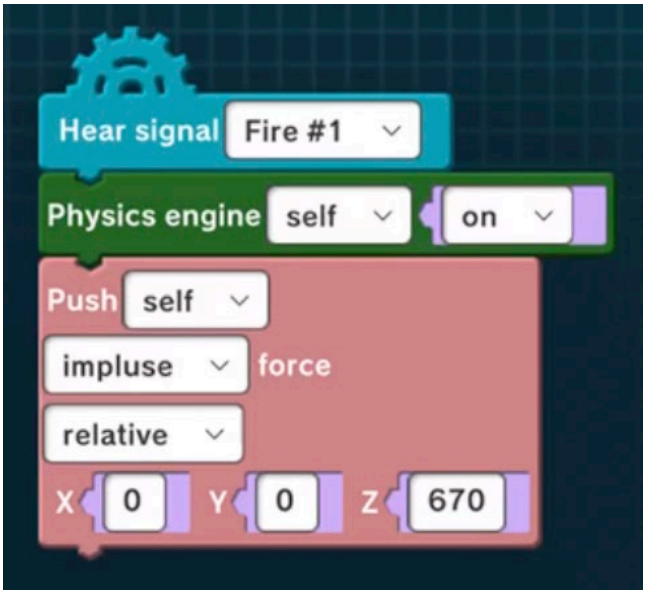

To send and receive a signal, use the two commands in the "Event" category.。




The program on the left is for controlling the overall flow. In this sample, a "fire button" is added, and tapping this button starts the entire process. By sending the signals in sequence, the program alternates between "loading" and "firing". It takes 8 or 15 seconds after sending each signal. This is because signals will be sent one after another unless the command "Wait for seconds" is inserted, and for example, the shell will be fired even though it is not set. Try and decide for yourself how many seconds to wait.




When a signal such as "Load!" or "Fire #1" is sent, the program on the receiving side is executed. The left figure shows the process when a "Load!" signal is received. The program from Lab “Three Targets” is used.

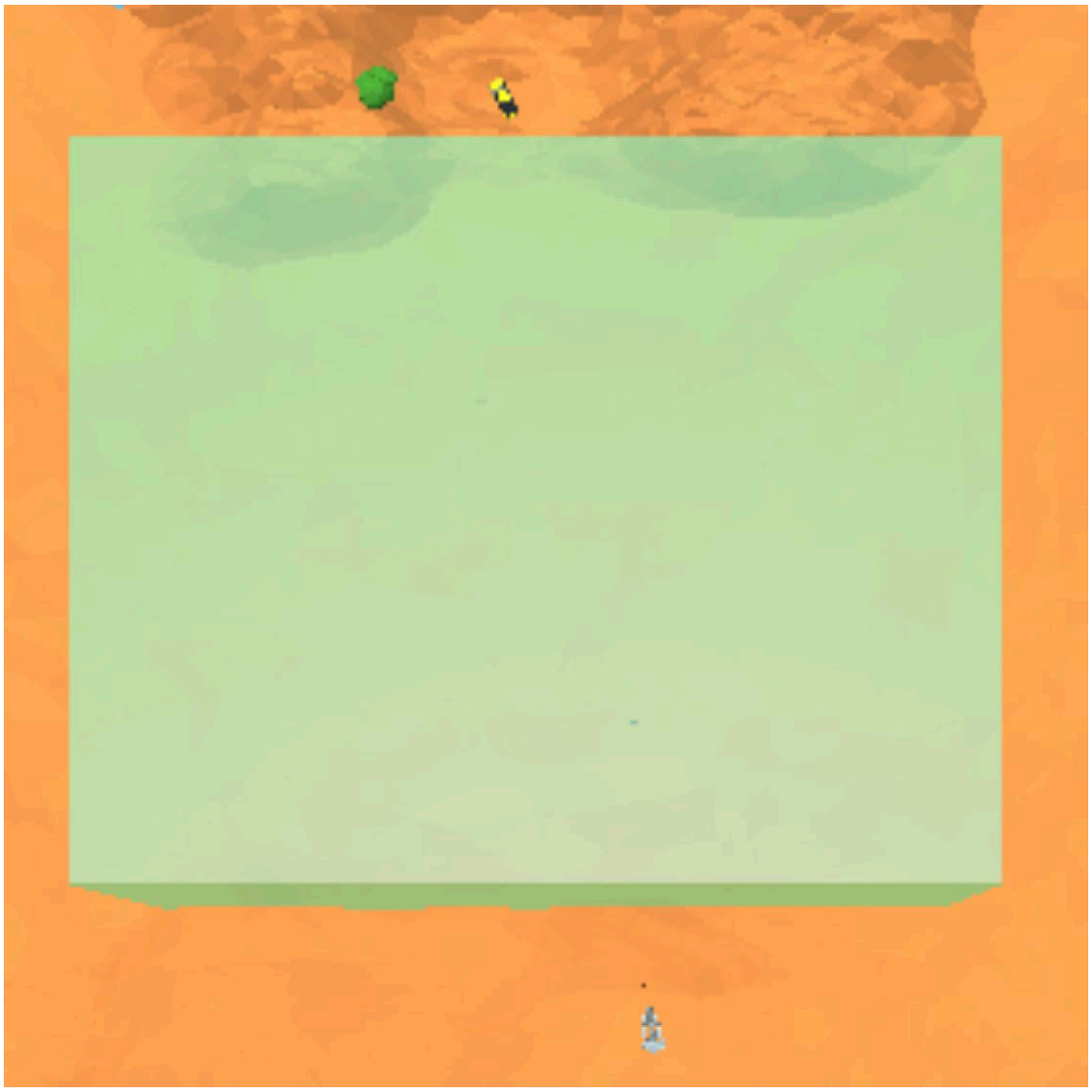



The program in the left figure also uses the one from the lab "Three Targets", and only the command at the top is changed to "Hear signal Fire #1". Fire #2 and Fire #3 are also made by modifying the program of the lab "Three Targets" in the same way.

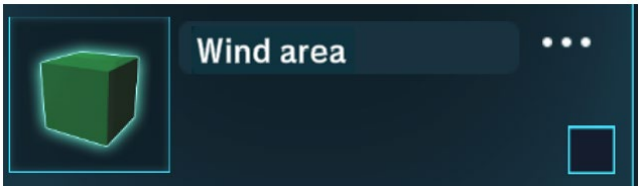
In Lab "Three Targets", there are programs that are executed when the A button or B button is pressed, so you will need to change them when making your own. Be careful.

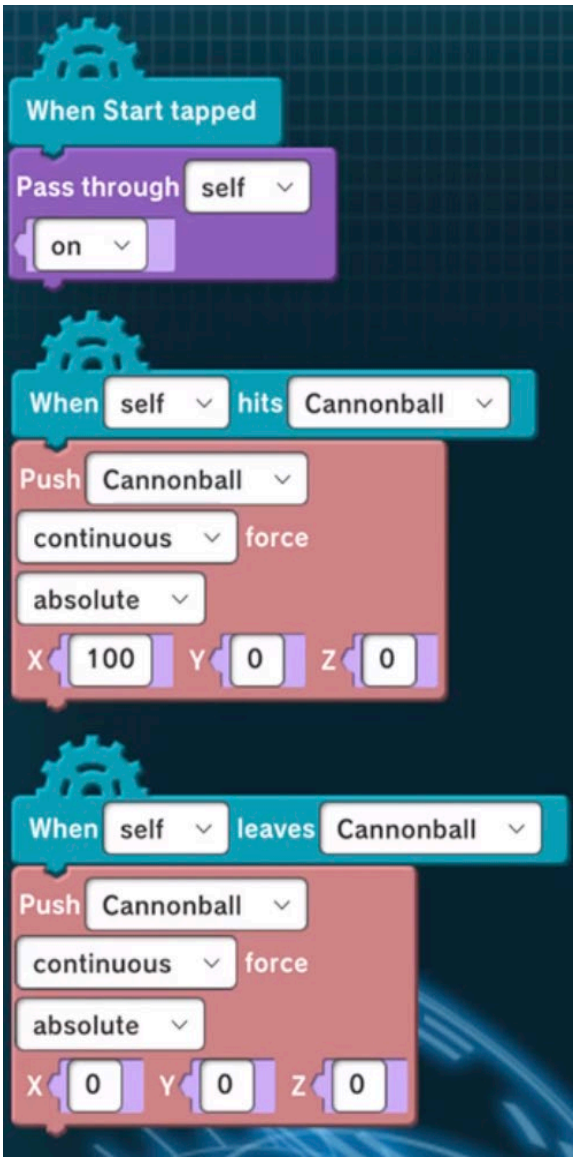
## Crosswinds Impact

In the Lab "Crosswind Impact", crosswinds blow while the cannonball is flying. The area where the crosswind blows is indicated by a green box for easy understanding. The inside of the box is the range where the wind blows. When a cannonball is fired and passes through it, the crosswind pushes the cannonball and affects its trajectory. You can't hit the target unless you investigate what kind of forces are working and how you can adjust it.



In this lab, an object called "Wind area" has been added. Inside this object is a program that makes the wind blow.



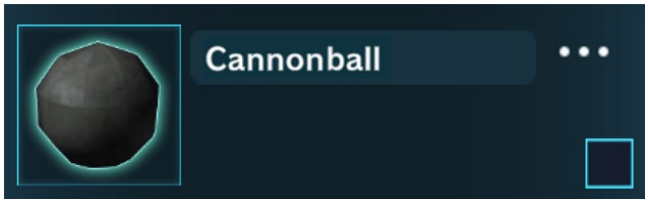



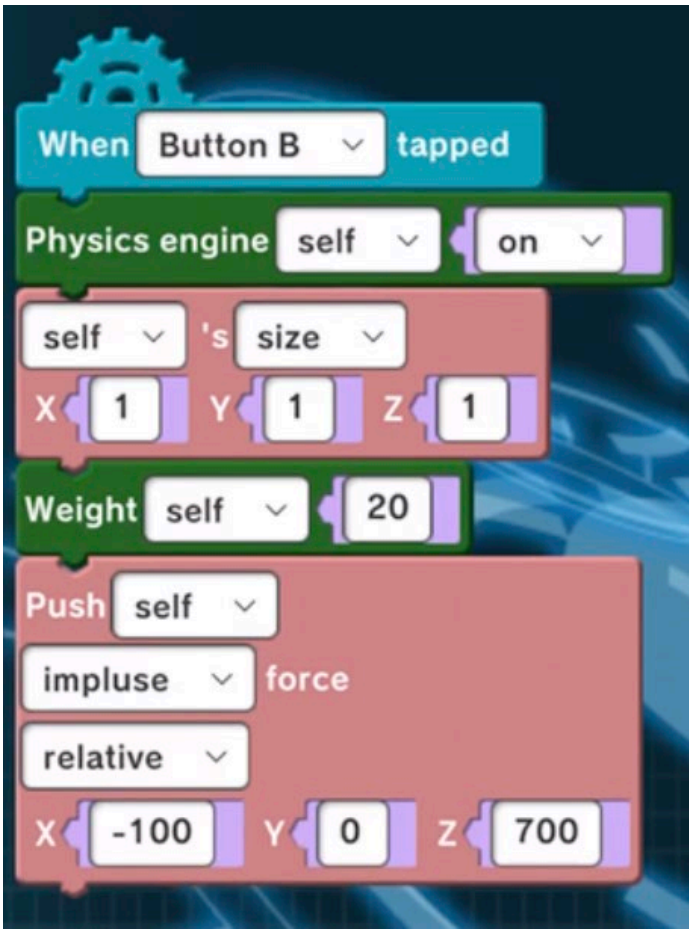

The left figure is the program of the Wind area object. The first program is the setting after the start button is pressed, and the "Pass through" is turned on so that the cannonball can pass through the green box.

The second program applies wind force to the cannonball from where it hits the green box until it reaches the other side. Because of the "Push Cannonball with continuous force", as long as the cannonball is inside the green box, it will be pushed by a force of size 100 in the x direction (horizontal direction).

The third program is to stop applying continuous force when the cannonball passes through the green box. The force applied in the x direction becomes 0。

In the program for the cannonball object, the X-direction force (100), which is continuously applied in the green box, must be counteracted in order to hit the target.




In the sample program, as shown in the left figure, a force of -100 is applied in the x direction by impulse force to self (cannonball). However, this setting will cause the cannonball to deviate too far to the right of the target.
The force applied to the cannonball is an impulse force, a force applied only at the moment of firing. On the other hand, the force applied inside the green box is a continuous force that adds 100 magnitudes in the x direction from where the cannonball touches the green box until it passes all the way through the green box. 100 in the x direction means that the force is applied from left to right when looking at the target from the cannon barrel. The cannonball gradually bends to the right in the green box.

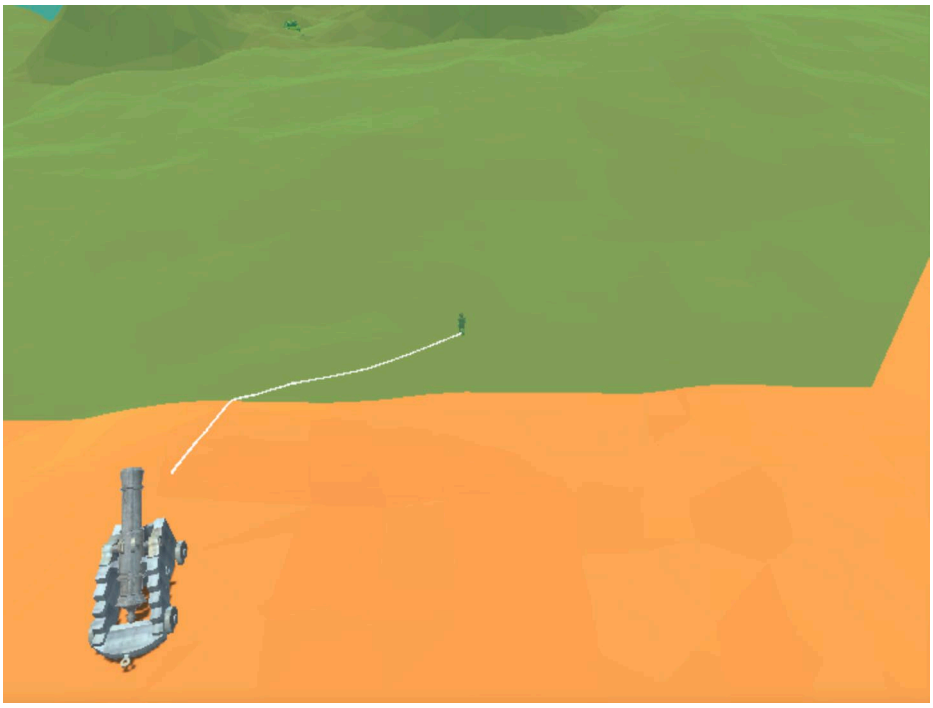
Apply a negative impulse force in the x direction to counteract the force applied inside the green box. Find out by trial and error how much force (eg -100) you need to apply to hit the target.

The direction and magnitude of the force applied to the cannonball can be freely changed by changing the program of the "Wind area" object.

Let's try blowing various winds such as headwind, tailwind, wind blowing from below, and wind blowing from above. Once that's done, let's see how we can hit the target when the wind blows from different directions.

## Advanced Challenge

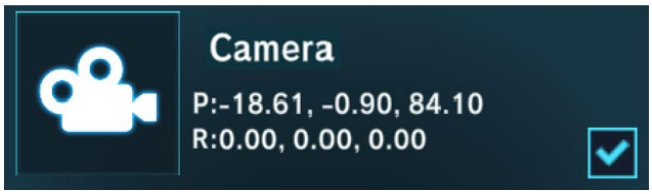
Try moving things other than cannonballs in the wind to see how they are affected. In the example below, we added a "Rogue" from the object menu and let him walk in the wind area. Let's try with various other things such as balls and cars.

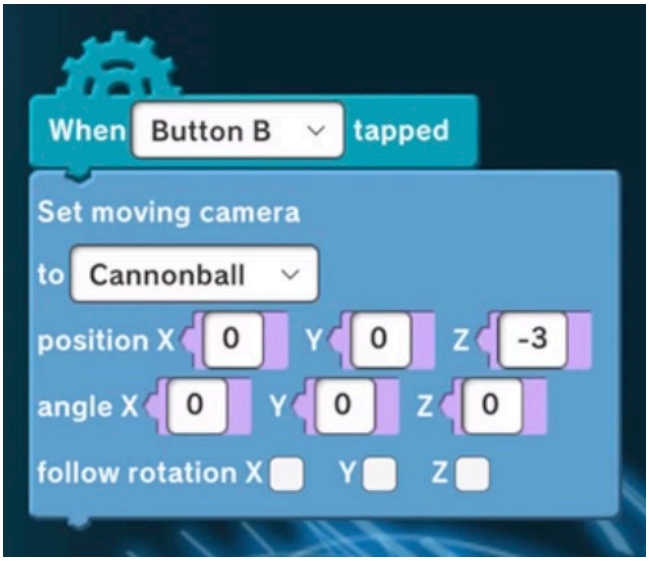


# Shoot a Cannonball 2

Lab "Shoot a Cannonball 2" puts the camera very close to the cannonball and follows it as it flies. It feels like you are on a drone chasing a cannonball and flying from cannon to target. You can use Mind Render's VR feature to make the experience even more compelling.

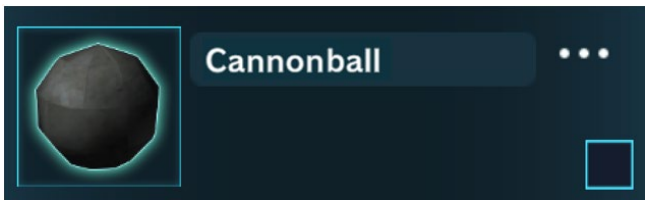

In this Lab, the camera settings are important. Take a look at the program for the camera object.

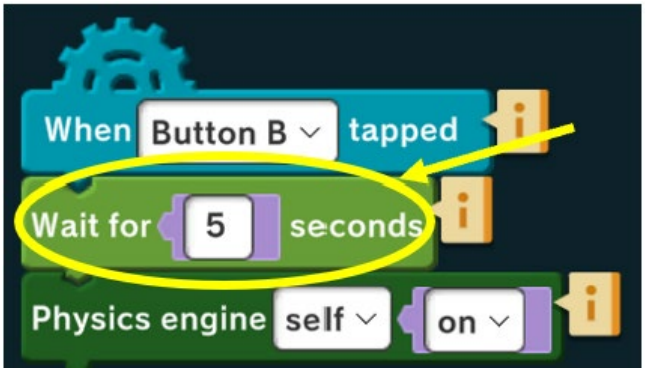





As soon as button B is tapped and the cannonball is fired, the camera is attached to the cannonball. The camera is mounted at the z direction (-3), slightly behind the cannonball.

The sample program of this lab is devised for enjoying in VR. You can see that by looking at the program for the cannonball object.

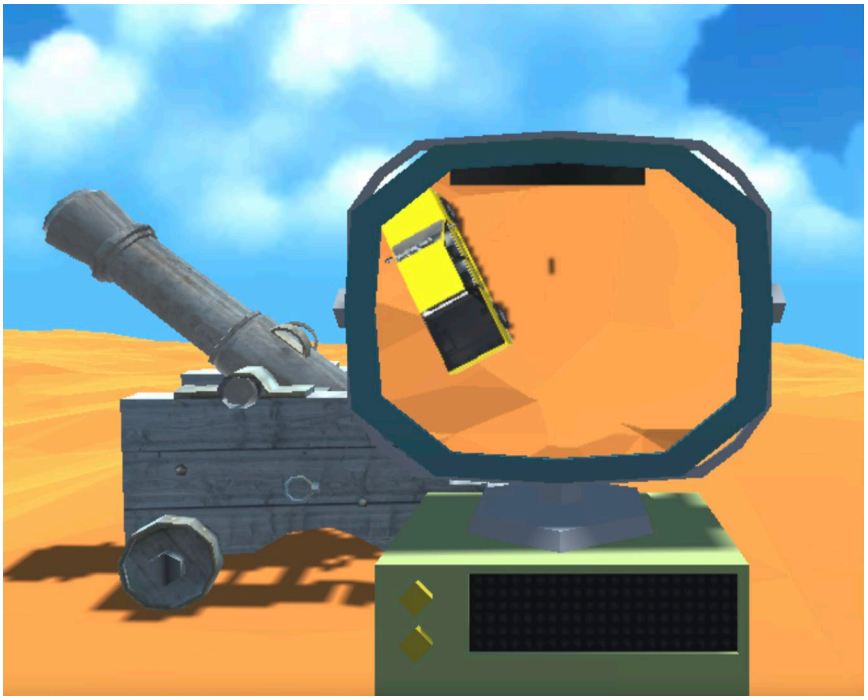



The command block to shoot the cannonball is the same as in the "Shoot a Cannonball" lab, but it waits 5 seconds for the cannonball to fire after button B is pressed. This is the time to prepare for VR, such as wearing VR glasses. If you don't play VR, you can remove this command.

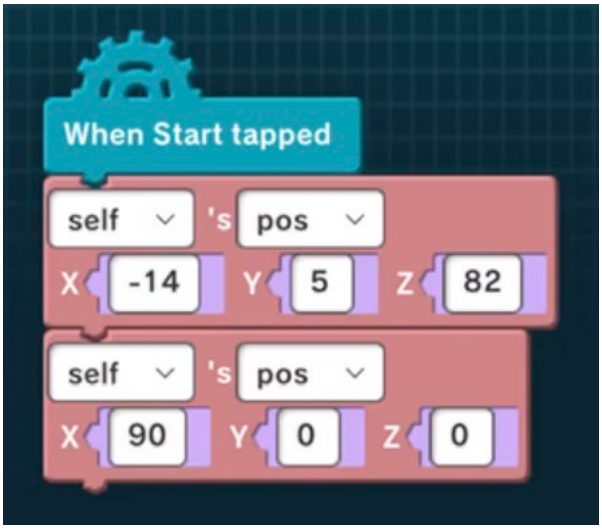

Please enjoy powerful 3D images and VR experiences by devising the mounting position of the camera.

## Advanced Challenge

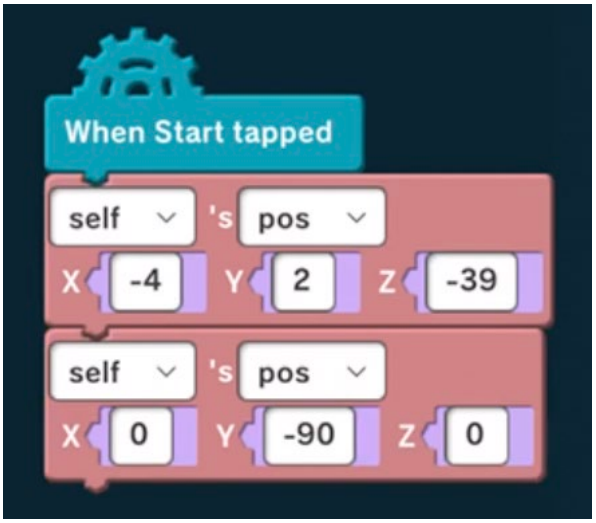


Object menu -> Model -> Program -> "Sub camera" and "Sub monitor" can be used to observe the far away situation from a distance. Using this mechanism, let's place a sub-monitor at the location of the cannon and see how it hits the target. Below is some helpful information.


A sub-camera and a sub-monitor are required to see the state of a remote place. Find and add it from the object menu.




The position of the sub-camera can be set freely. In the example on the left, the angle is set so that it faces straight down on the scrap car that is the target.

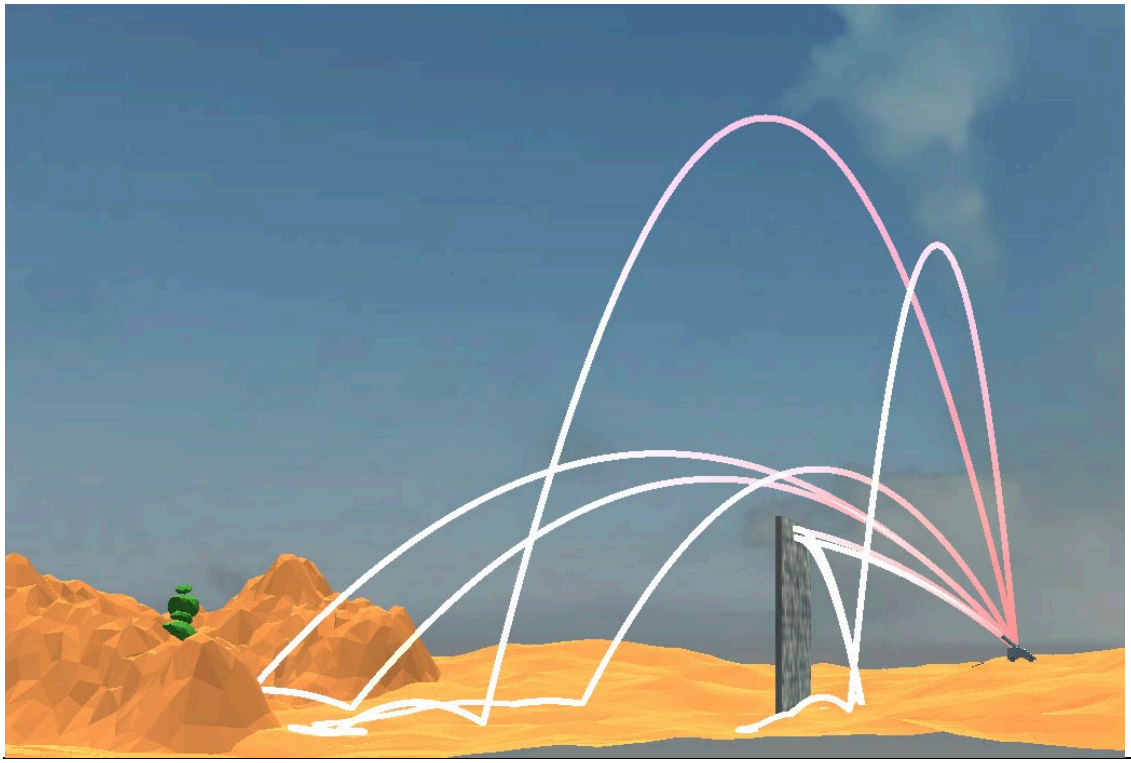



In the left figure, the sub-monitor is placed next to the cannon, and the angle is set so that the screen can be viewed from the side.

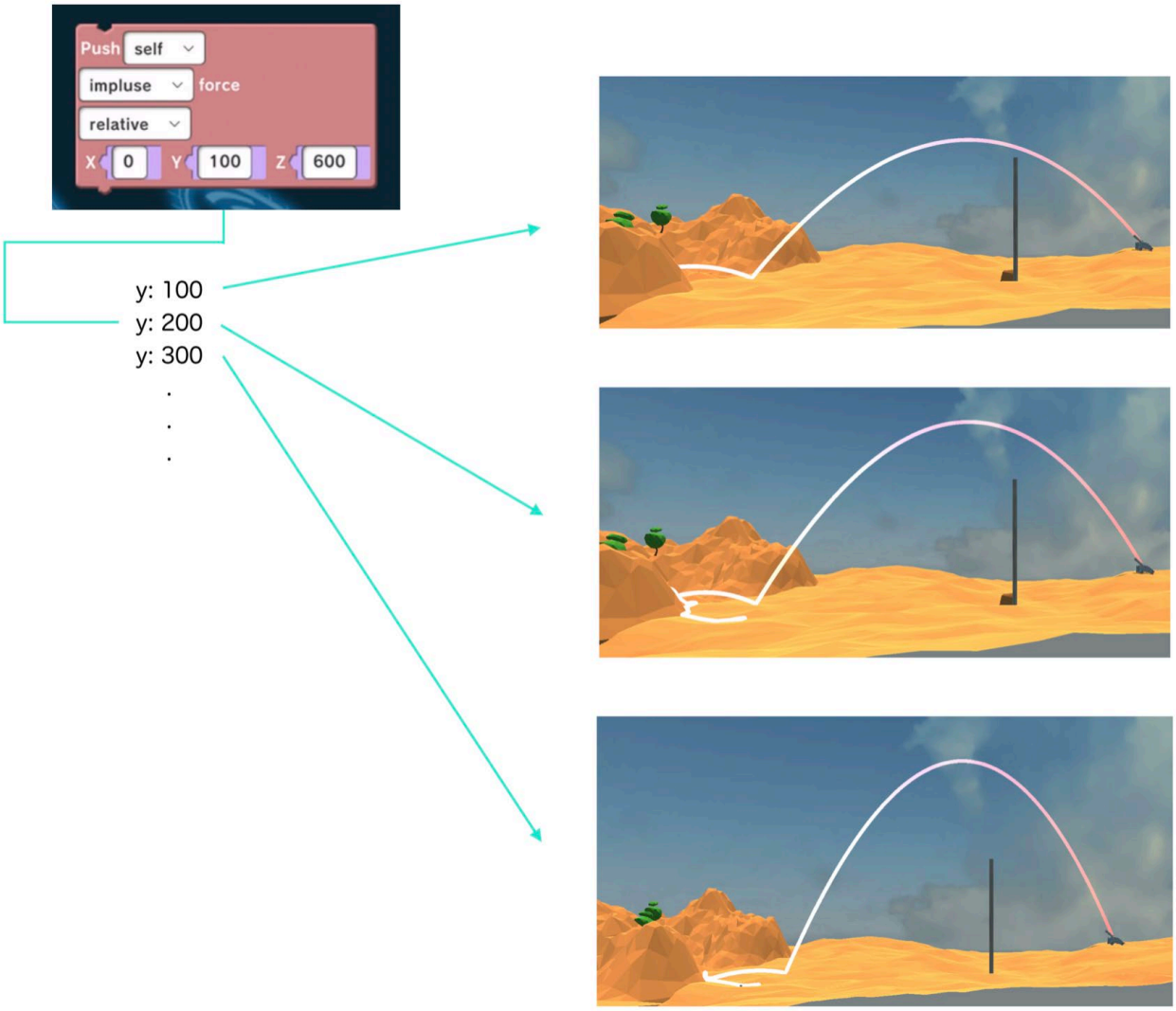
## High Wall

In Lab "High Wall", a large wall will appear between the cannon and the target. Hit the target over this wall.

Try different ways to change the trajectory of the cannonball to hit the target over the wall.

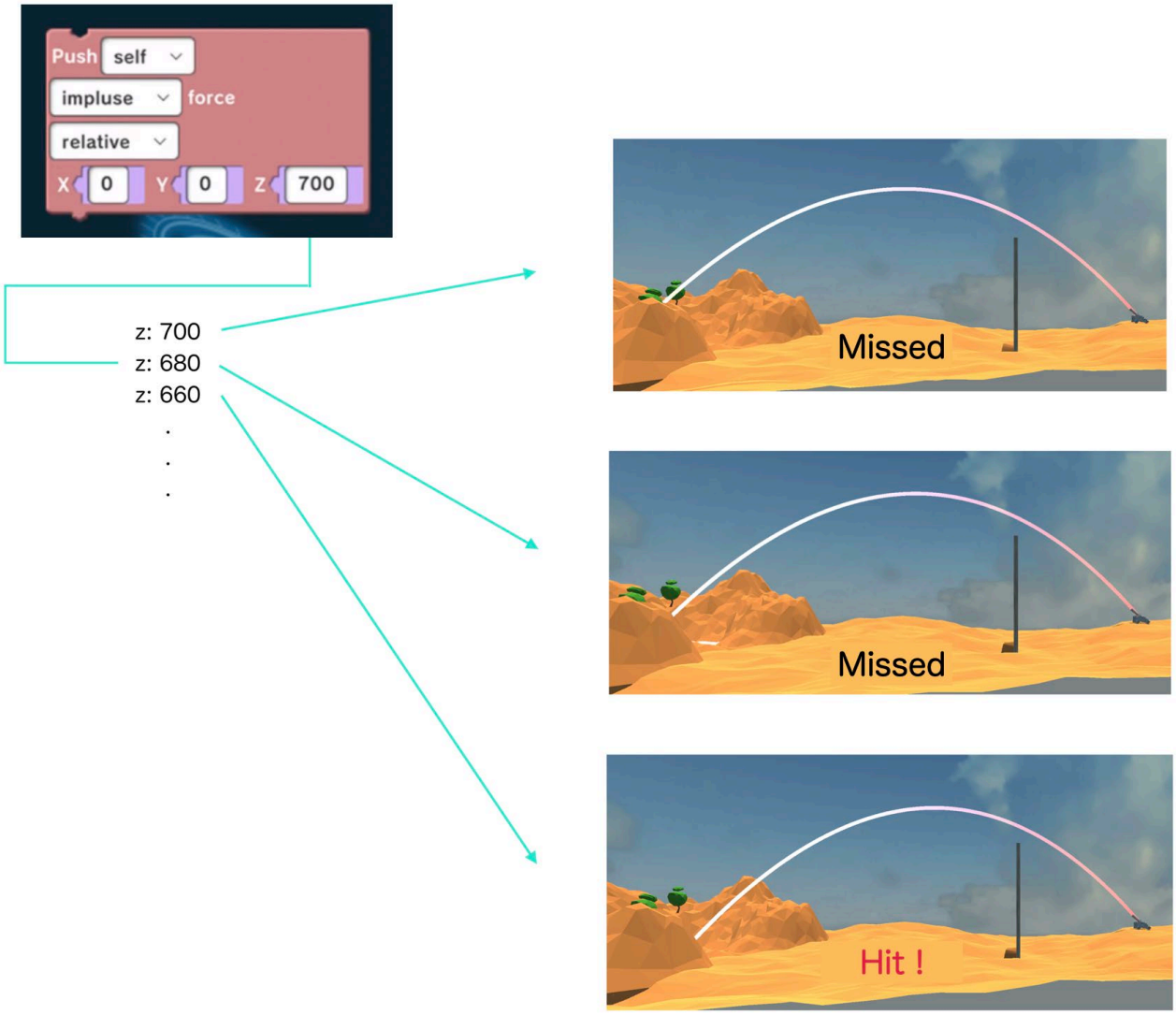


If you leave the sample program as it is, the height of the trajectory of the cannonball is not enough, so it will hit the wall. Let's try variously changing the force in the vertical direction with the impulse force when launching.



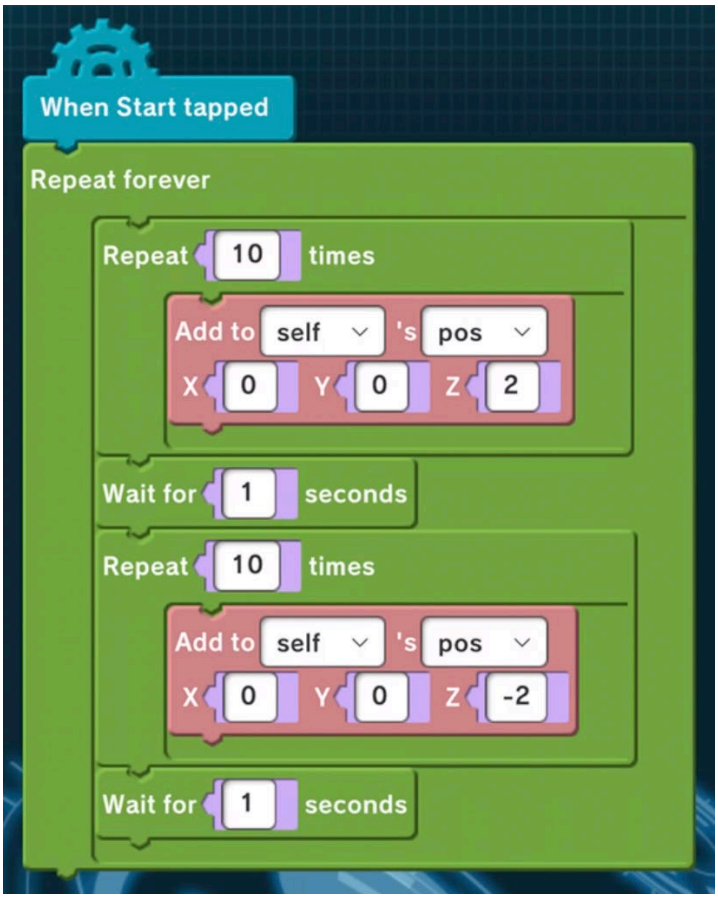

As the force applied in the vertical direction increases, the height at which the cannonball launches increases, but the reach distance decreases. It seems necessary to increase the force applied in the traveling direction (z-axis direction) at the same time as launching high.

Along with increasing the force applied in the vertical direction (y), adjust the force applied in the direction of travel (z) to find a setting that hits the target.



## Advanced Challenge

Think about what you can do to make it more interesting and add programs. As a reference example, the program that moves the wall back and forth is introduced below.



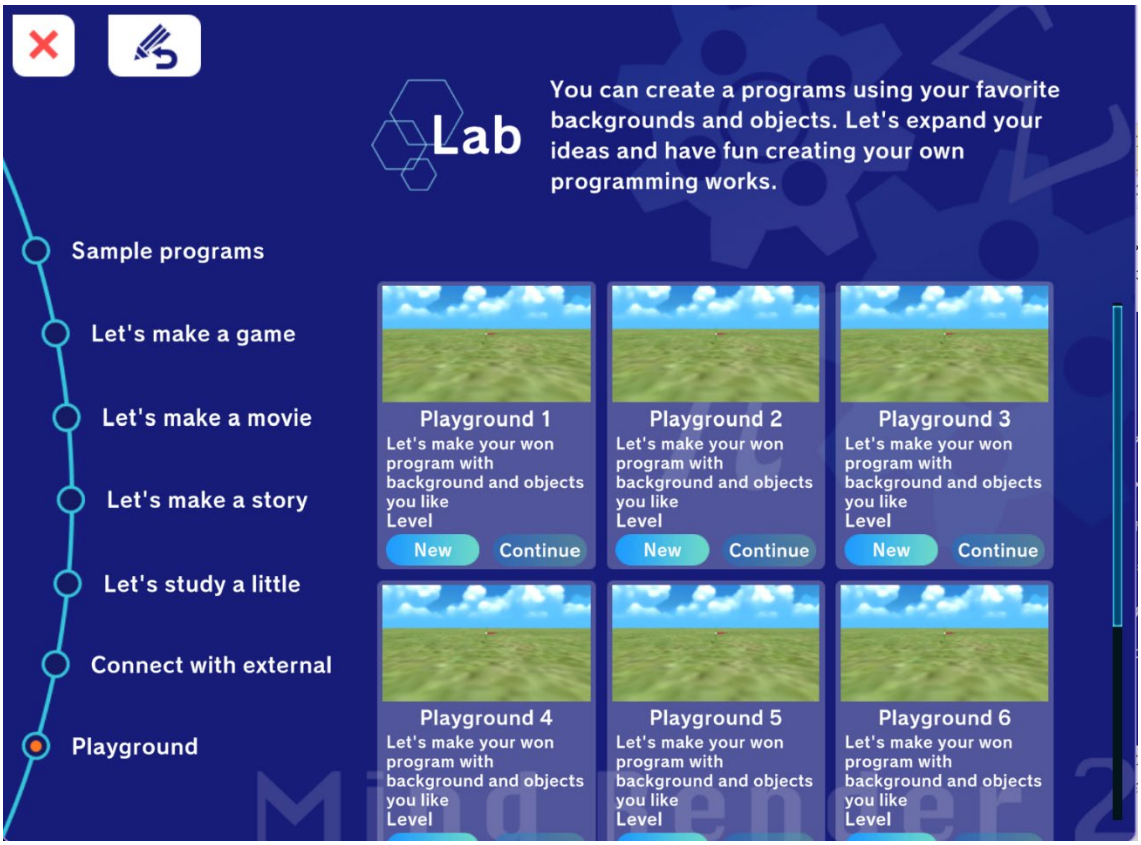

The wall will move closer or further away from the cannon. Adjust the difficulty of the game by changing the speed it moves.

Try other ideas, such as automatically raising or lowering walls.

# 12. Create your own work using Playground

## Challenge to become a movie director and make a short movie!

By selecting "Playground" from the lab menu, you can freely create your own work. There are 1 to 9 playgrounds, so you can make up to 9 projects. If you want to make more works, you can also save them in the locker. For how to use the locker, please refer to "Tips for using Mind Render 2" at the end of this document.



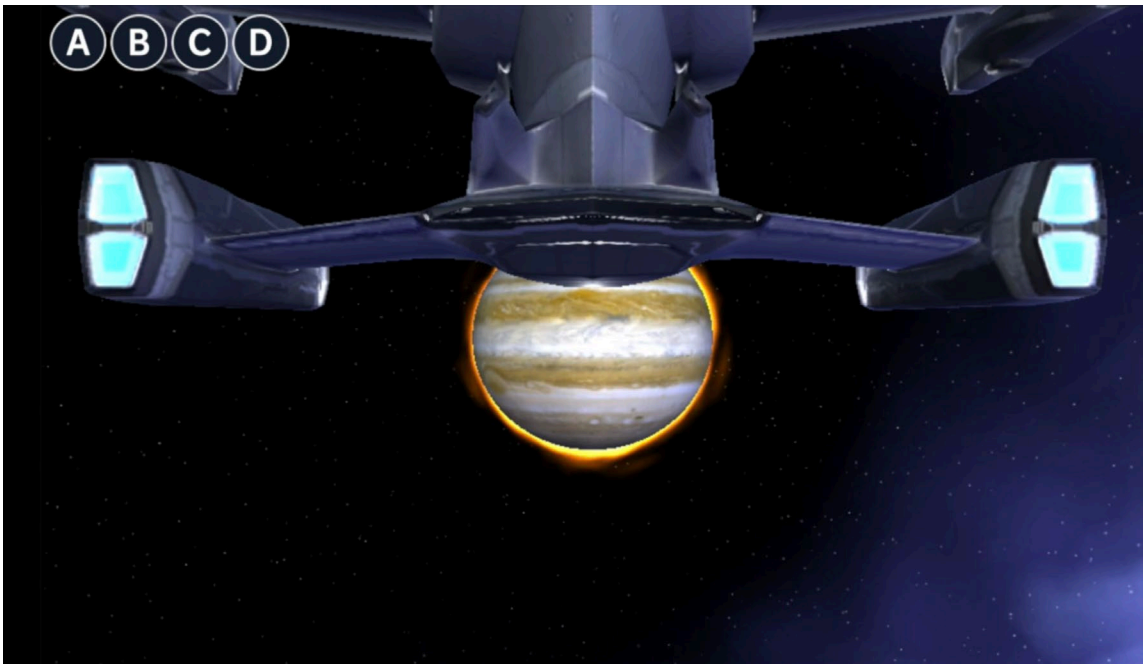

## How to make a short movie

Here, we will introduce how to make a short movie as a hint for how to use the playground.

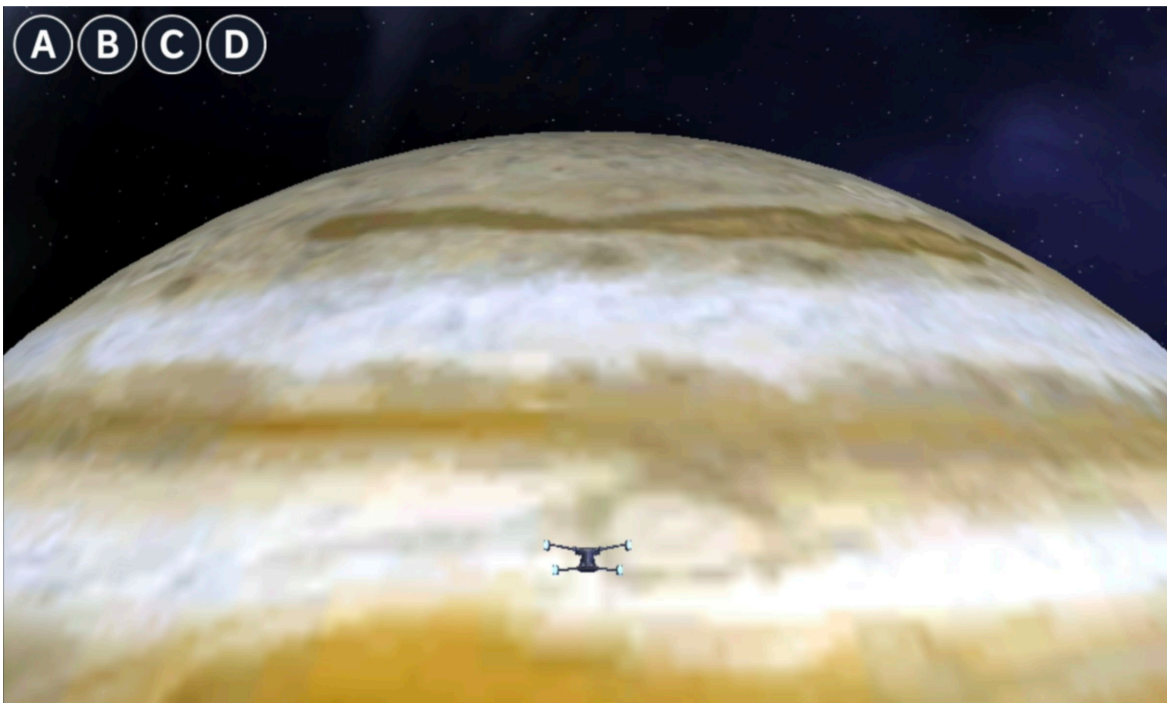
* You can see the completed program by Lab > What do you want to make? > Let's make a movie.

Mind Render provides commands to use a fixed camera or a moving camera freely. By controlling the coordinates, angles, and movements of the camera, you can create powerful images like movies.

Example 1 ) You can display how the spaceship flies toward the planet (using the Jupiter object) and how big the spaceship is by setting the camera at an angle that looks up from below. By slowing down the movement of the spaceship, you can create a more impressive scene.




Example 2) A spaceship is approaching the planet and entering the atmosphere. The camera coordinates can also be set inside the spaceship, so you can create a scene where you can see the appearance of a huge planet from the cockpit.



Example 3) By adjusting the coordinates, angles, and movements of the camera and objects, it is possible to capture in the same scene how a distant spaceship grows in size as it approaches, and how other objects such as robots move simultaneously.

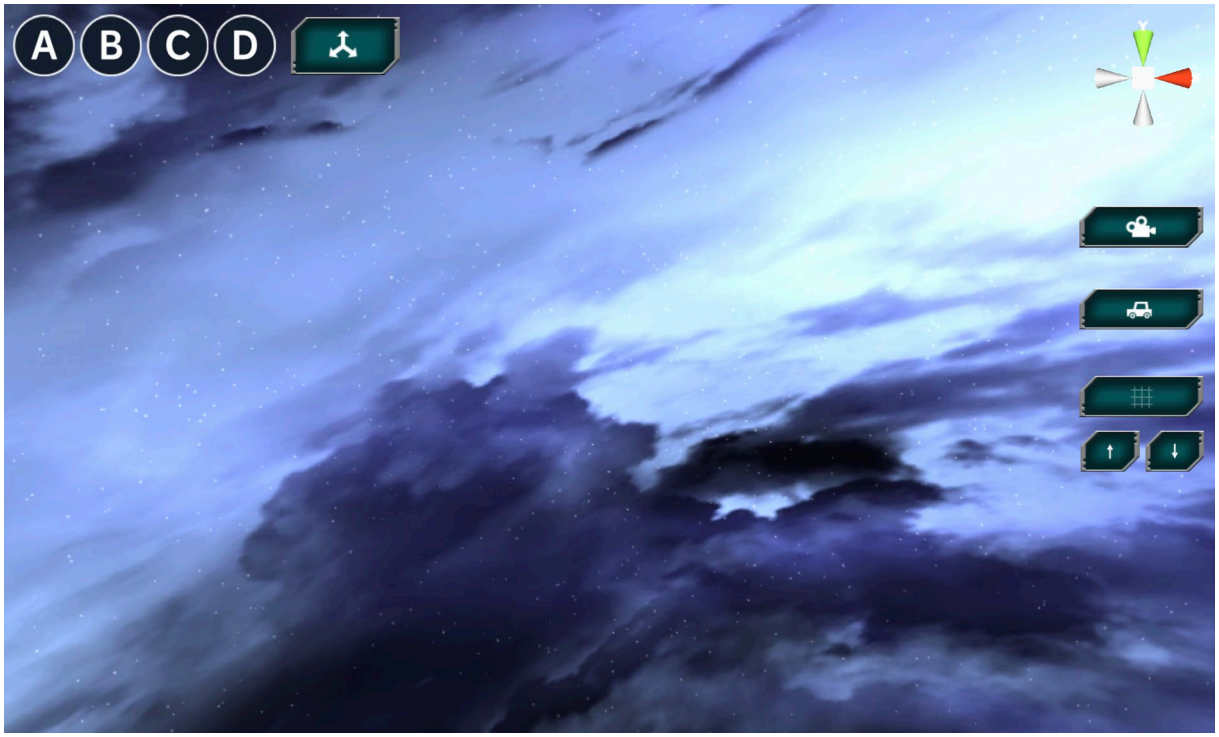

Example 4) This is a scene where the camera approaches standing characters from behind.

To create a scene that you often see in movies, consider the movement of each frame and adjust the position between the character and the camera. To further enhance the visual effect, you can also add effective sounds. Mind Render provides various sounds useful for movie making such as "Horror", "Action", "Battle" And "Fantasy". Furthermore, by combining sound effects such as "Explosion", "Crowd", and "Beam", you can create more realistic expressions.

Next, let's see what kind of programming is required to create a specific scene. The next page introduces an example of a 30-second short video.
The program is included in "Free Research 5: Short Movie". Please refer to that as well.

First, we will introduce the flow of the video with five screenshots of Example 1). After that, we will look at the sample program in Example 1).

Scene 1) The camera captures a certain direction in the wide universe. From here, the camera is slowly lowered toward the bottom of the screen. Cosmic atmospheric sound is heard in the background.



Scene 2) The spaceship eventually comes into view from the bottom of the screen. The spaceship is moving toward the back of the screen, but the camera goes under the spaceship and look up to show the bottom of the spaceship.

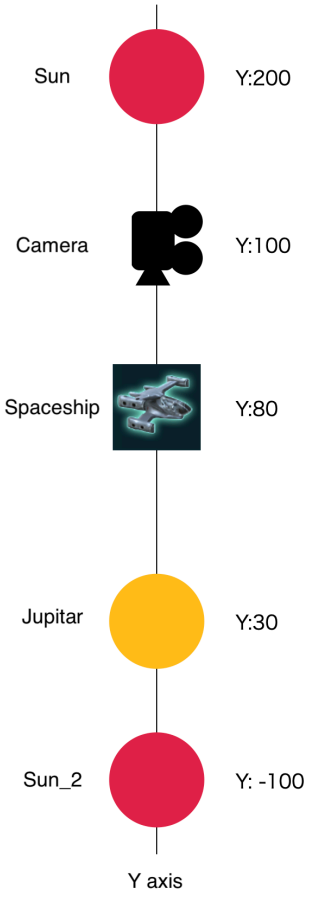

Scene 3) The camera moves toward the planet in front, showing the bottom of the spaceship. The bottom of the spaceship will be slowly projected.

Scene 4) The camera projects to the forefront of the spaceship and moves forward, overtaking the spaceship.


Scene 5) Soon, only the planet will appear on the screen, and eerie music will play as if something is about to begin.

Screen captures may not be enough to convey the atmosphere of the video. The sample programs from Example 1) to Example 3) are explained below, try making your own and see how it goes.

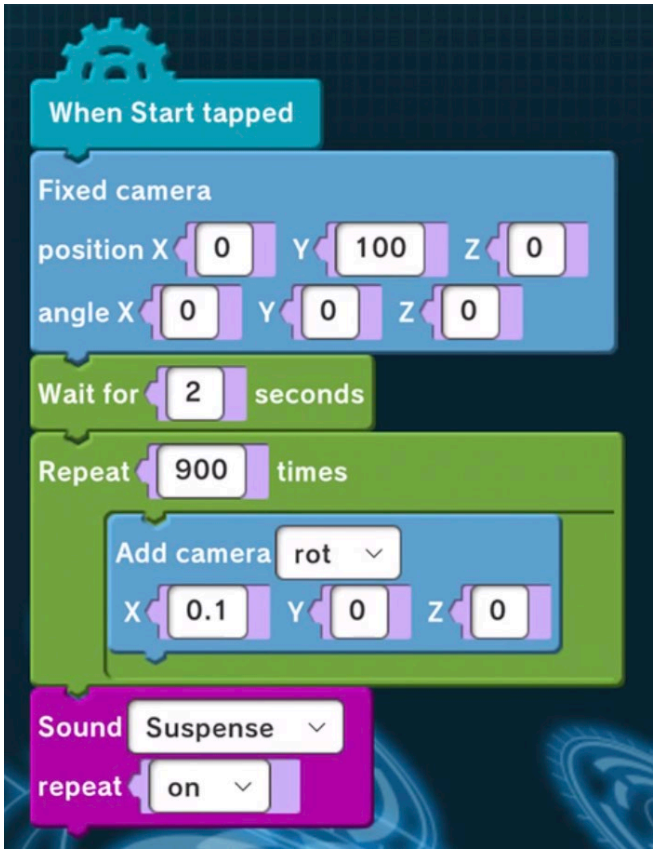
First, the program for the Example 1) will be explained below.
Below is a list of the objects used to create example 1).


To move objects along the flow of this scene, set the initial coordinates of each object as shown in the left figure. All objects are aligned on the Y axis. The initial values of the X and Z coordinates are all 0. Place Sun 2 behind Jupiter and use the solar flare to shine around Jupiter.

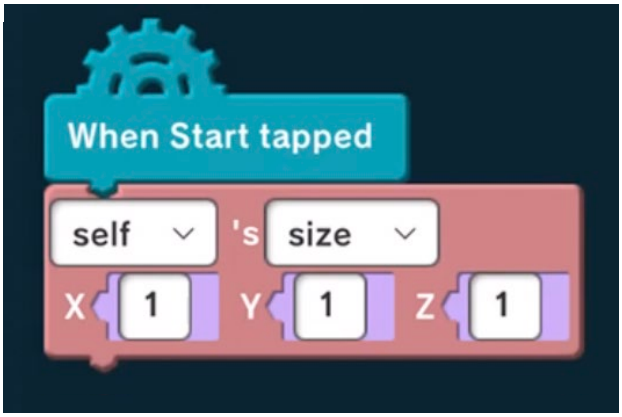
The sun is placed at Y: 200 coordinate to brighten the surface of Jupiter. Without this sun, the surface of Jupiter would be dark, and the patterns would not be visible.

Let's look at the program for each object in order from the next page.



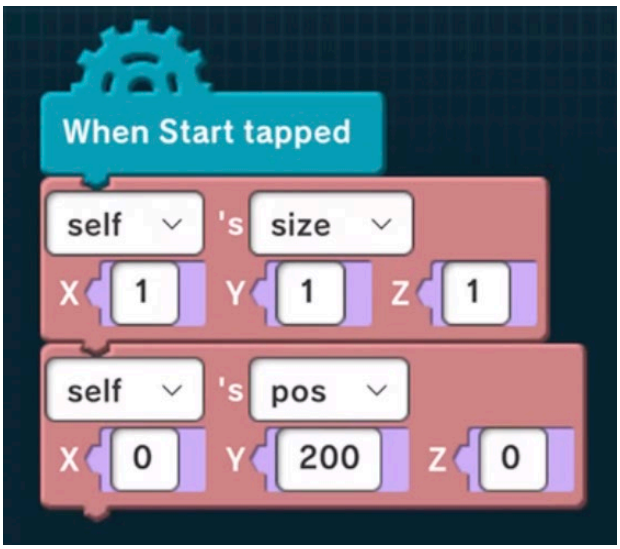

Camera object

The left figure shows the program for the camera object.
As soon as the start button is tapped, the "Space" sound will be played. Set the loop to "on" so that it repeats over and over.
The coordinates of the fixed camera are X(0), Y(100), Z(0), and the angle of the fixed camera is all set to 0.
After waiting for 2 seconds, rotate the camera slowly along the X axis. When the rotation of the camera is over, start playing "Suspense" sound. Set the loop to “on” for this sound as well. "Space" and "Suspense" sound will be played at the same time.



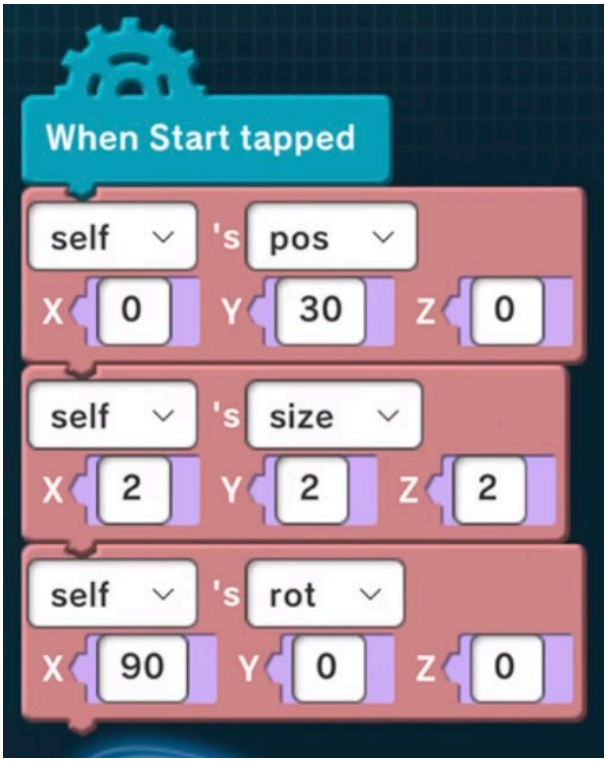

NewTown (background)

The default playground background is NewTown. Set this to "Universe (blue)" by replacing objects.



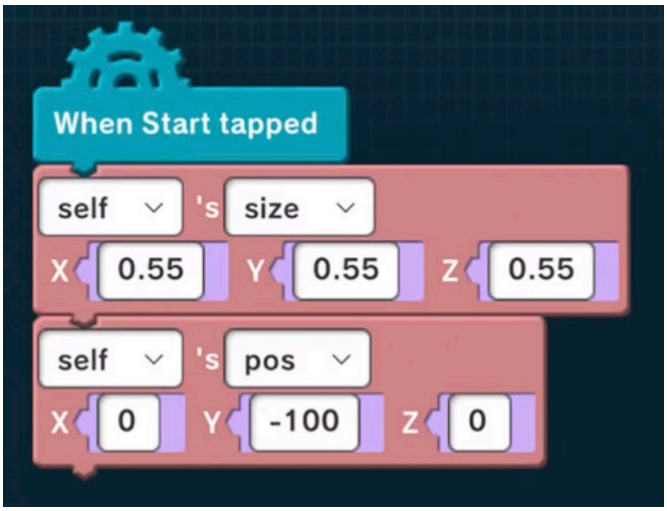

Sun

Set the size and coordinates of the sun as shown on the left.



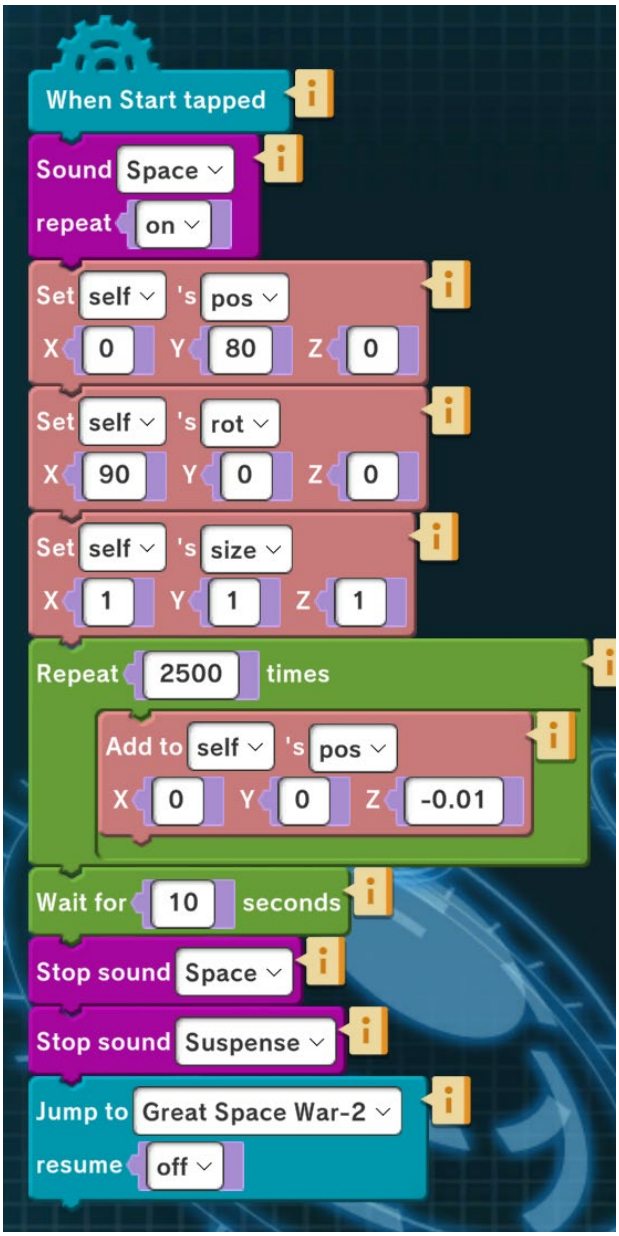

Jupiter

Set the coordinates and size of Jupiter as shown on the left. Jupiter looks different depending on the viewing angle, so adjust the X coordinate position so that the side you want to show faces the camera.




Sun_2

Adjust the size of Sun_2 so that it does not protrude from Jupiter. The coordinates are shown on the left.



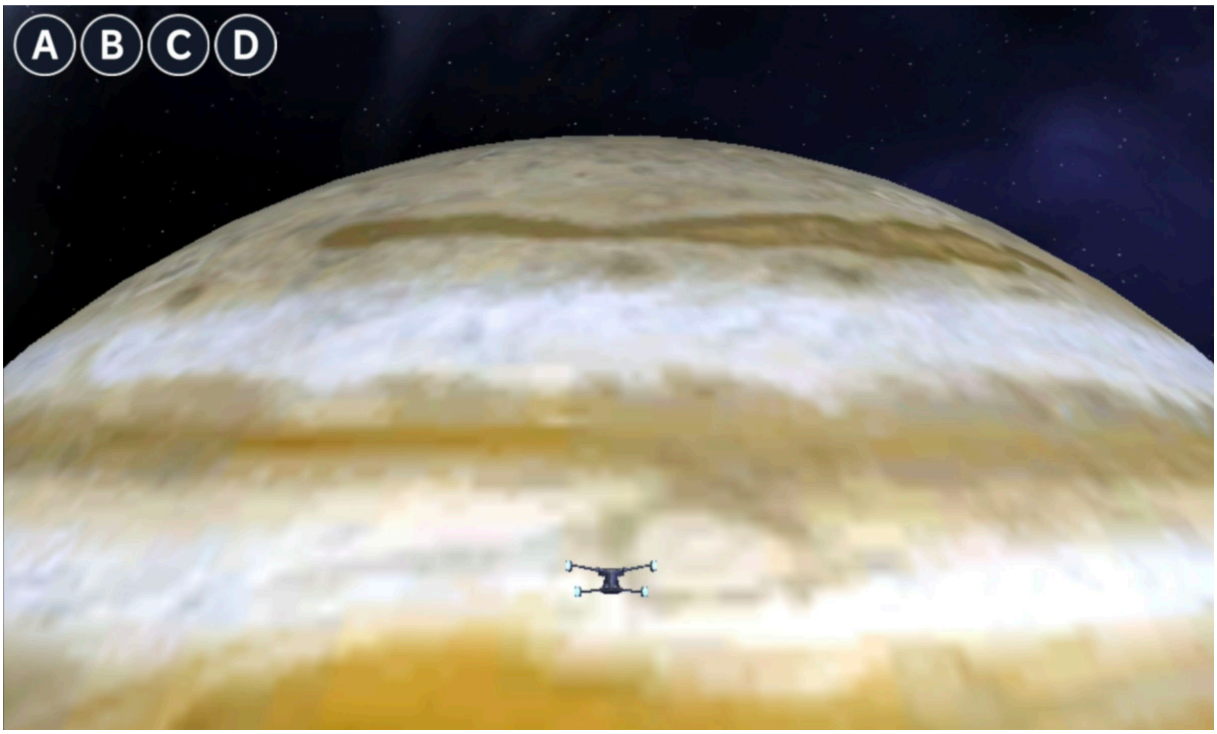

Spaceship

The initial coordinates of the spaceship are shown in the left figure. The angle is rotated 90 degrees in the X-axis direction so that the tip faces Jupiter. By decrementing the coordinate value of the spaceship's Z-axis direction (travelling direction) by 0.01, the spaceship is made to slowly retreat above the fixed camera. Wait 10 seconds after the repetition is done so that you can hear the sound while looking at Jupiter.

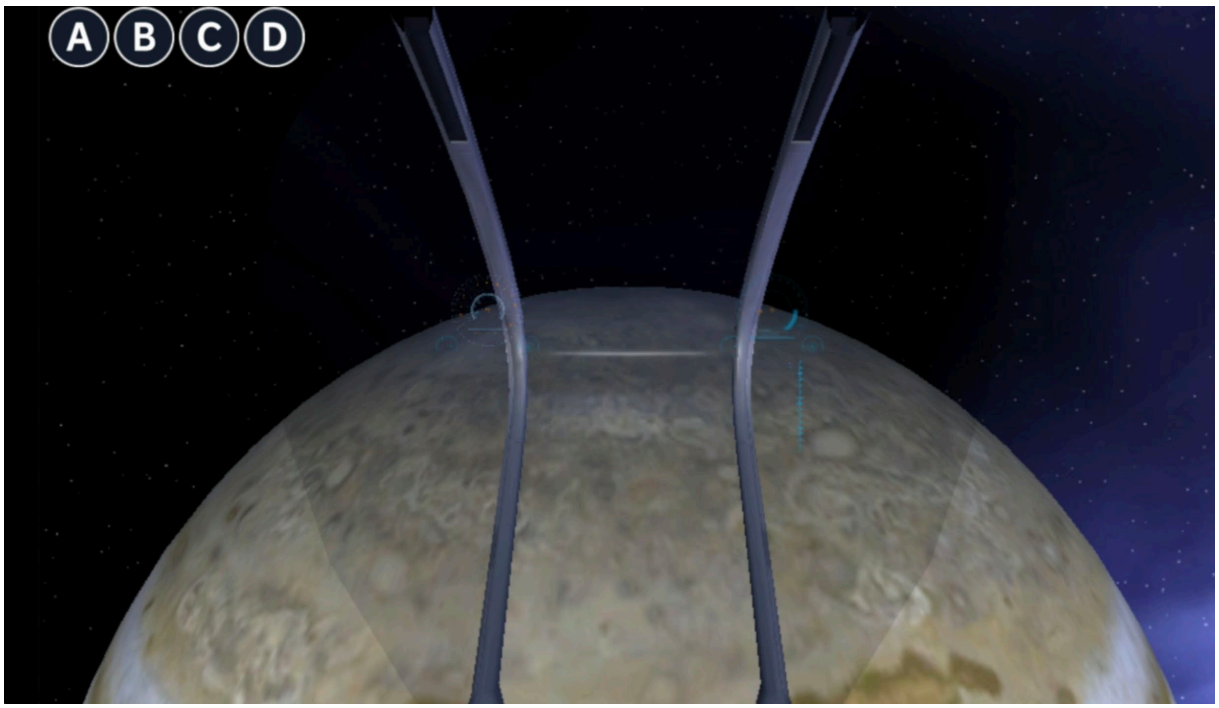
Finally, jump to Great Space War-2 to switch to the next scene. The reason for jumping to another program is because there can only be one background at a time, so if you want to change the background you have to create a separate program.

Next, we will introduce the scene of Example 2).

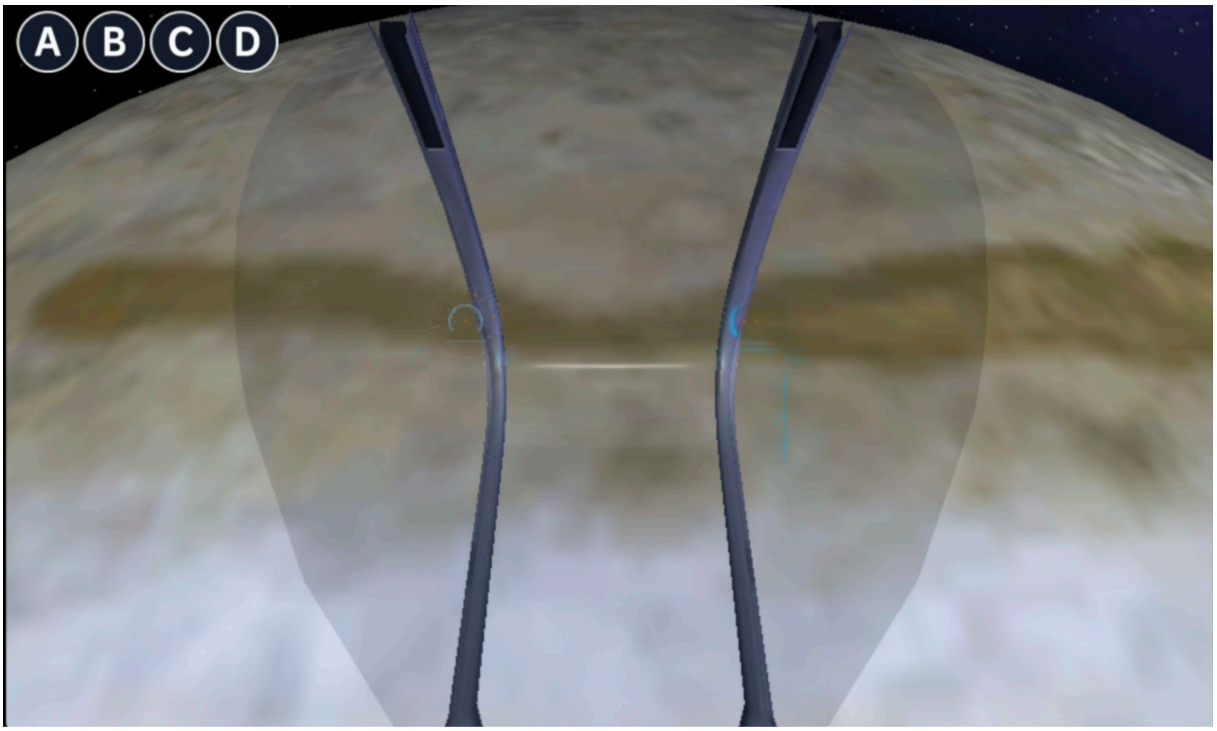
Scene 1) is a scene where you can see a spaceship approaching Jupiter from afar. By making the spaceship look small and Jupiter look big, it gives the impression of the largeness of the planet.




Scene 2) View from the cockpit of the spacecraft. It is a scene where we gradually approach Jupiter. Jupiter appears larger and larger and is rotating at the same time.

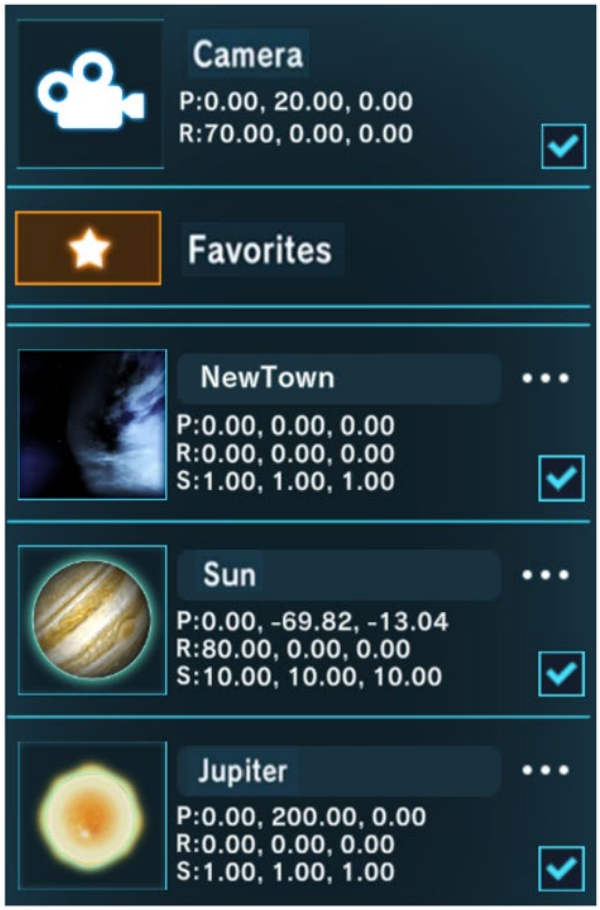


Scene 3) The spacecraft plunges into Jupiter's atmosphere.



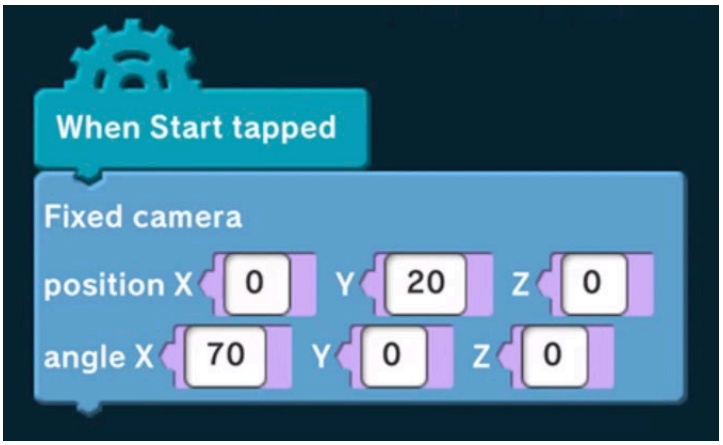

Scene 4) State of entering Jupiter's atmosphere. The visibility is getting worse.

Below is a list of objects used in Example 2).



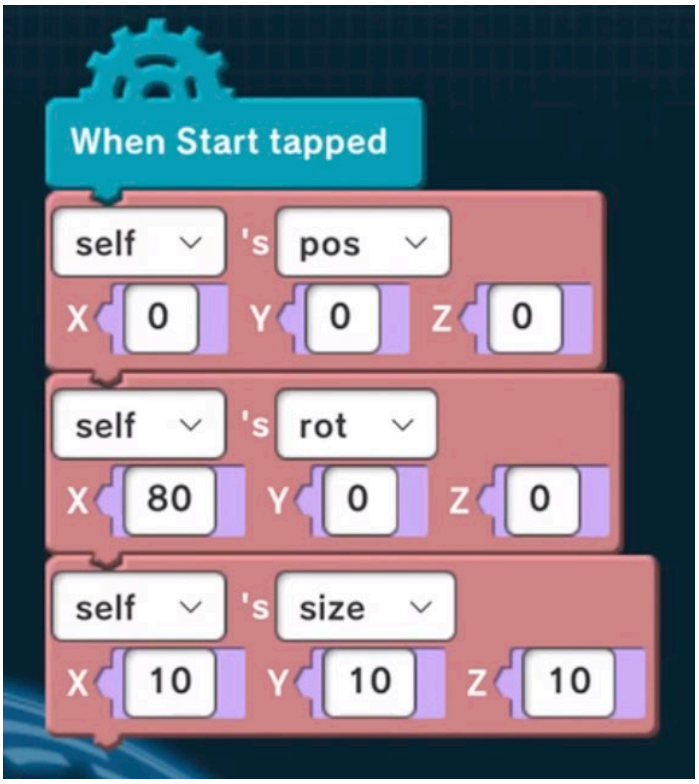


From the next page, the program of Example 2) will be explained for each object.



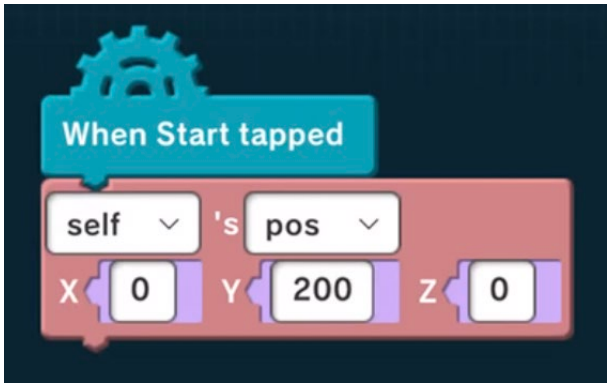

Camera

The initial coordinates and angles of the camera are as shown in the left figure. It is a setting that Jupiter is captured in front, and a small spaceship is visible between the camera and the planet.




Jupiter

The default settings for Jupiter are shown on the left. The size of Jupiter is emphasized by setting the size of Jupiter to 10 each in X, Y, and Z.




Sun

The Sun is placed behind the camera to illuminate Jupiter's surface so that the patterns on Jupiter's surface are brightly lit.


Spaceship

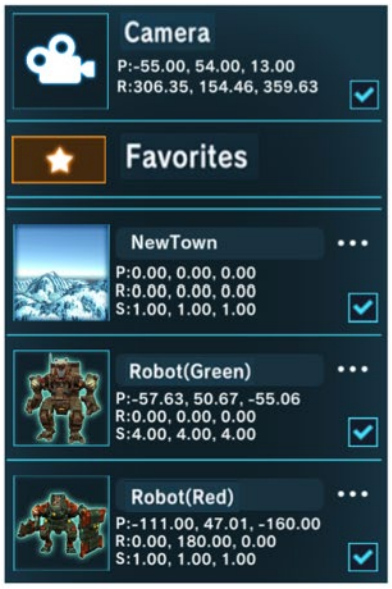
When started, play the “Space sound" and "Suspense" sounds with the loop setting "on". The initial settings for the coordinates, angle, and size of the spaceship are shown in the left figure. By setting the size as small as "0.2" just in front of Jupiter, you can see the contrast with Jupiter's size. Wait 3 seconds in this state, then change your coordinates to X(0), Y(35.19), Z(45). At the same time, let the coordinates of the fixed camera be X(0), Y(34.75), Z(45.5), and the angles of the fixed camera be X(90), Y(0), Z(0). These settings will move the camera to the cockpit of the spacecraft and display Jupiter as seen from the cockpit.
Once the spaceship and camera settings are set, gradually change the angle and coordinates of Jupiter to show the spaceship gradually approaching Jupiter. When visibility drops, wait for 5 seconds to give the impression that you are entering the atmosphere and moving through the atmosphere. The sound will continue to play during that time. After 5 seconds, stop the sound and jump to Free Research 5 -3 to move on to the next scene.

Next, we will introduce the scene of Example 3).

Scene 1) A spaceship appears over the planet and prepares to land. A large robot is walking in the lower right corner and in the back of the screen.

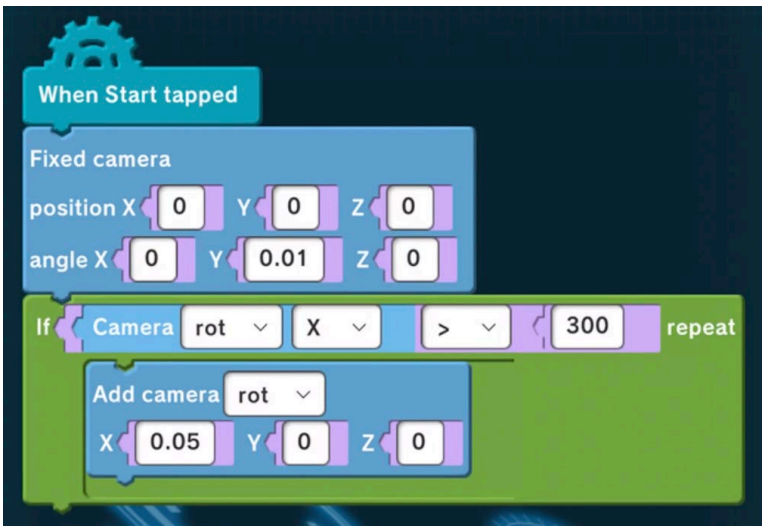

Scene 2) The spacecraft roars and lands in the landing space. A giant robot is passing by it.

The objects used in example 3) are as follows.






The program of Example 3) is explained below for each object.

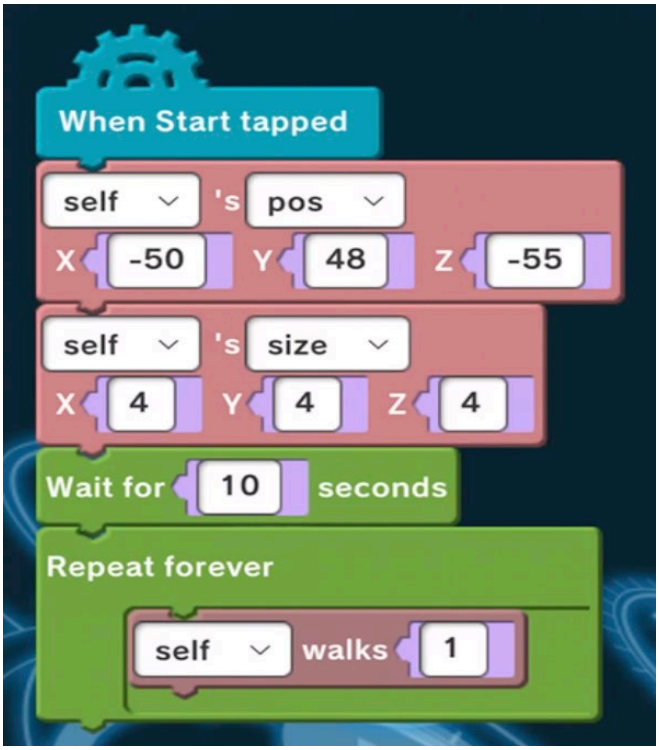



Camera

The fixed camera's coordinates and angles are set like this. After the initial setup, change the X and Y angles of the camera slightly to follow the spacecraft landing.

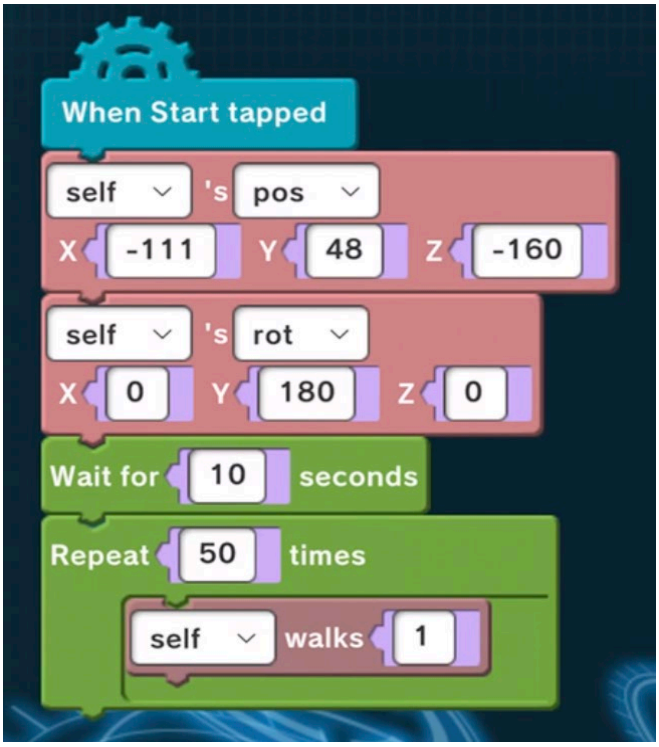
NewTown (background)

There is no program, change the background from the object menu to "Snowy mountain".




Robot (green)

Set the initial coordinates and size as shown in the left figure. These coordinates set the robot on the road right next to the landing site. The robot will wait until the spaceship comes down and then start walking.




Robot (red)

Set the initial coordinates and size as shown in the left figure. These coordinates will set the road ahead of the spaceship. The robot will wait for the spaceship to come down and then start walking.


Spaceship

After starting, the sound object "Car noise" is played repeatedly instead of the engine sound. The initial value of the coordinates is set in the sky as shown in the left figure.

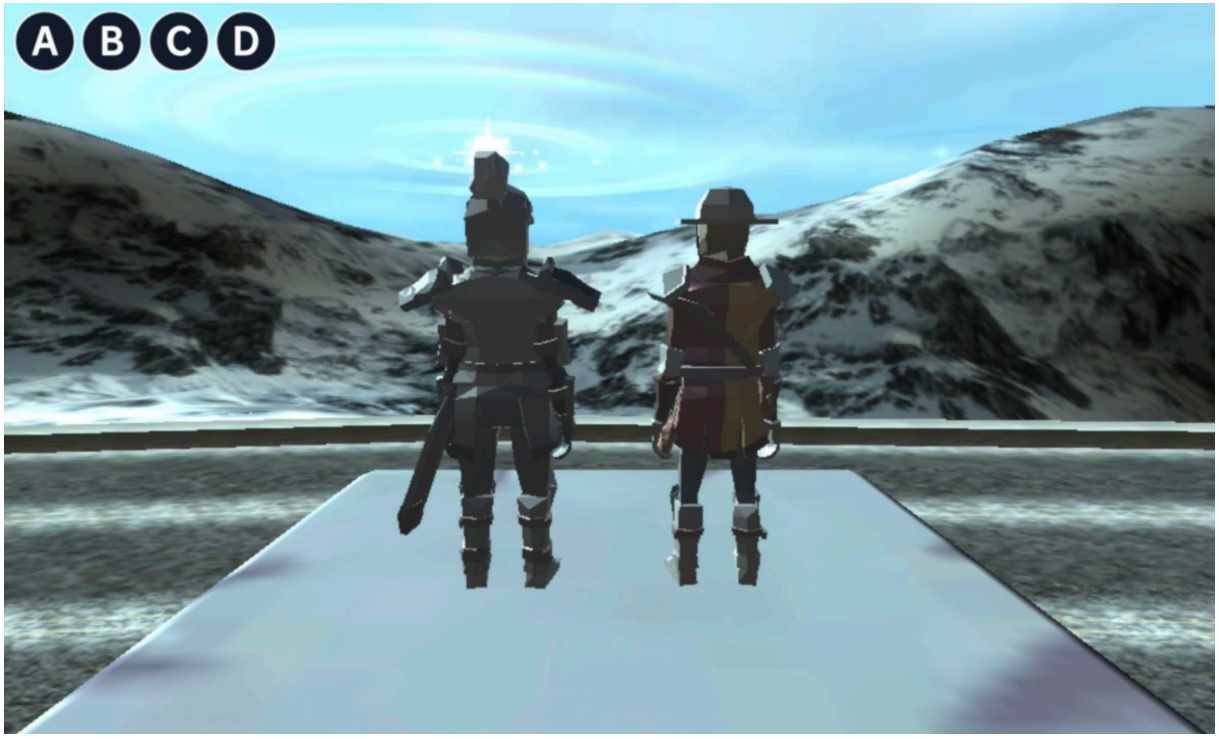
The state of gradually dropping altitude after entering the landing posture is expressed in two stages. The first 400 repetitions will have a slightly faster descent, and the next 100 repetitions will have a slower descent with smaller Y decrements. Also, the angle is slightly downward (Z-axis direction) as it approaches the landing position.

As soon as the spaceship lands, the engine sound ("Car noise") is stopped and the landing sound ("Scare") and the engine stop sound ("Power Down") are played.

Wait 5 seconds and jump to Free Research 5-4 to proceed to the next scene.

Next, we will introduce the scene of Example 4).

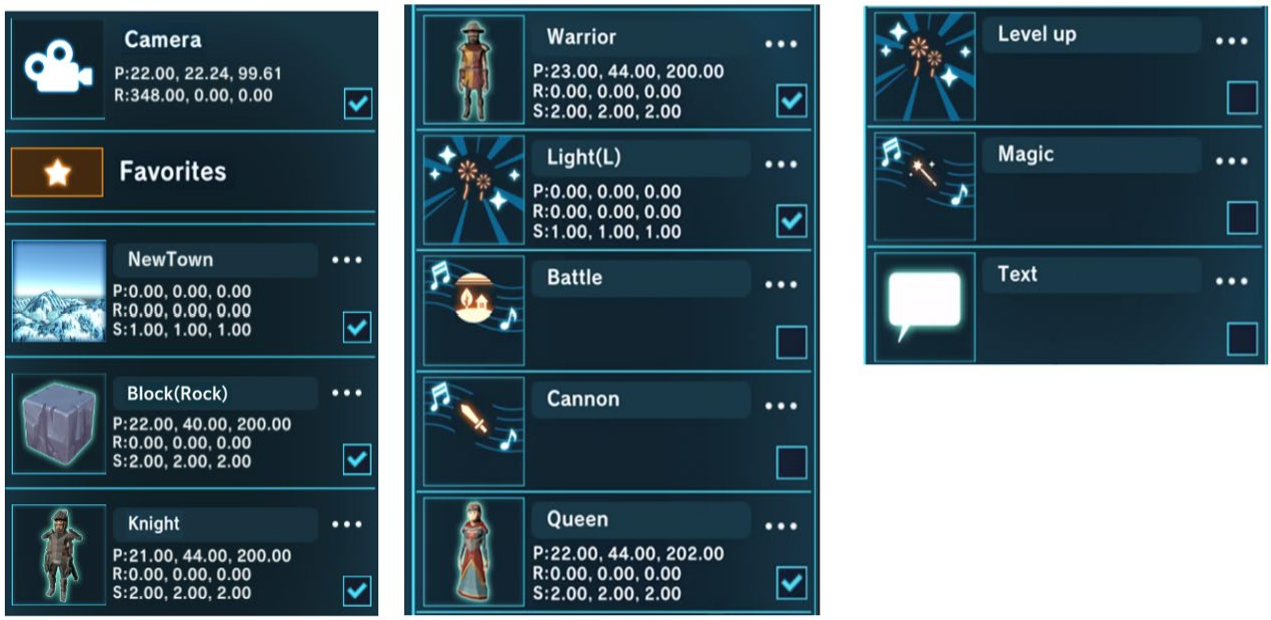
Scene 1) Two warriors are looking at the snowy mountains. Over the snowy mountains, in the distance, repeated flashes are seen along with loud explosions.

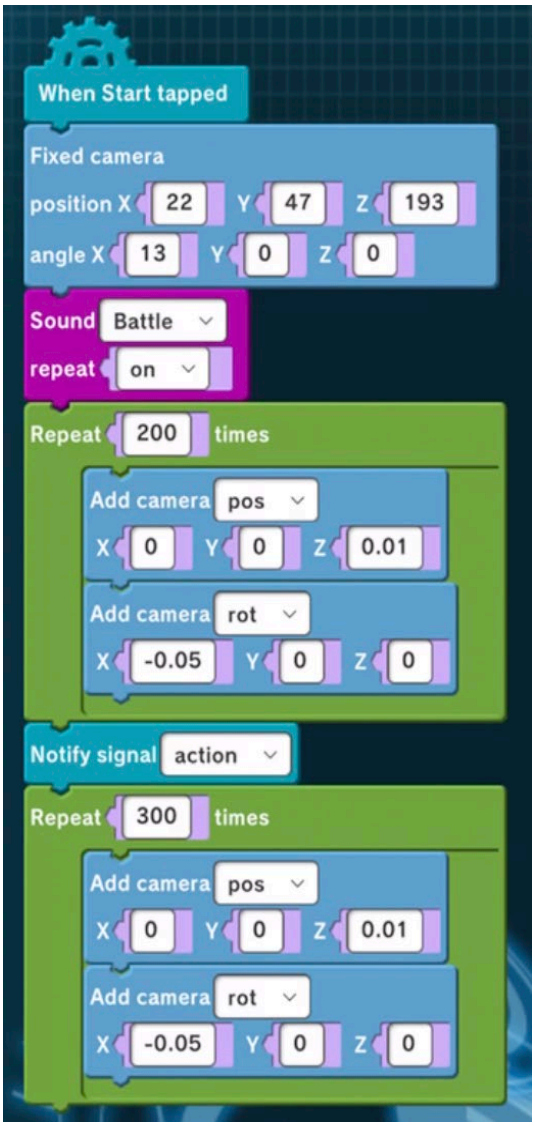



Scene 2) As the camera approaches the two warriors, the queen character appears as if she is warped with a flash of light.

Scene 3) Once the three characters get together, the camera approaches behind them and continues to capture the three characters staring at the distant battlefield (where the explosions and flashes are coming from).


The objects used in this scene are as follows. The program is explained for each object from the next page.

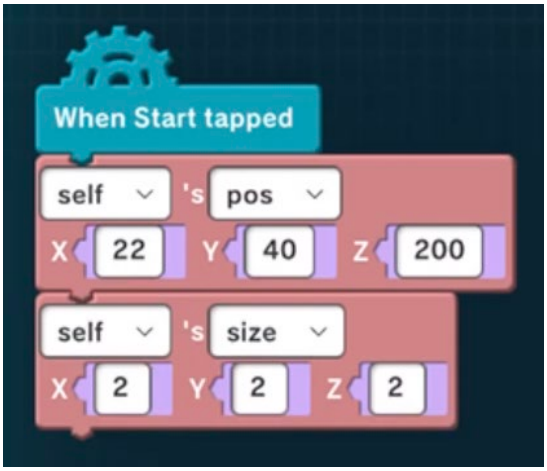





Camera

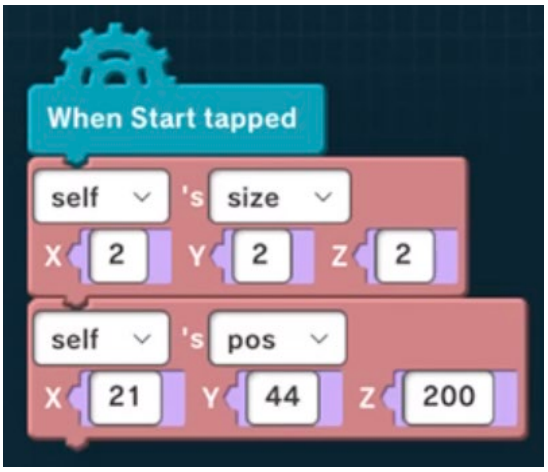
The initial settings of camera coordinates and angles are shown in the left figure. Put the snowy mountains into view from behind the two warriors. Simultaneously with the start, the sound "Battle" is played repeatedly.

Next, increase the Z-axis value of the camera by 0.01 to get closer to the warrior. At the same time, increase the camera angle by X (-0.05). By doing this, the camera will gradually move up as it gets closer to the warrior. After repeating this 200 times, send a signal called "Action". This signal is sent as a trigger for the Queen character to appear at this timing.
After sending the signal, move the camera closer to the warrior



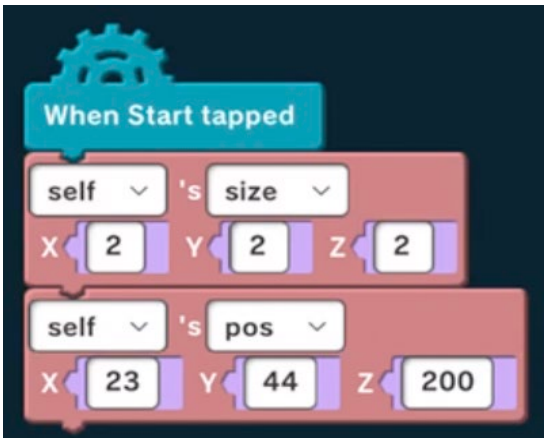

Block (rock)

The rock block will be used as a stage for two warriors to appear. The coordinate position and size of the rock block are as shown in the left figure.



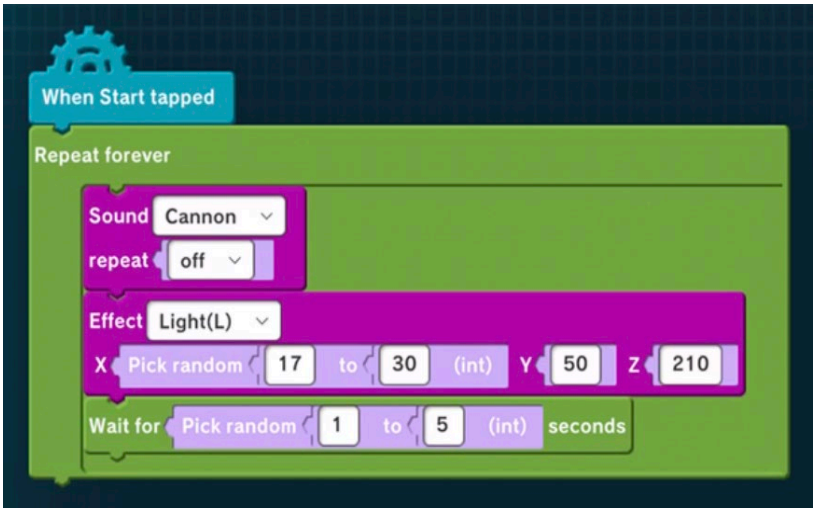

knight

The knight will be doubled in size in the X, Y and Z directions. Set the coordinates as shown in the left figure so that you can see him standing on the rock block.

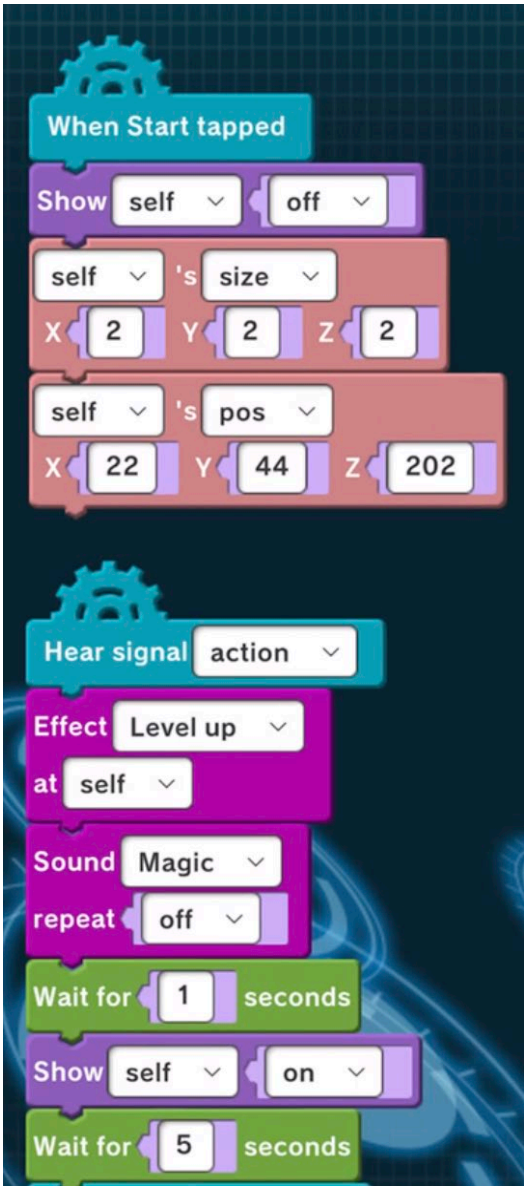



warrior

Make the warrior twice as large in the X, Y, and Z directions, like the knight. The coordinates are set as shown in the figure on the left so that the character stands right next to the knight.




Light (L)
The sound "Cannon" will be played along with of the effect. This effect is played repeatedly, but the value of the X coordinate is determined by a random number so that it will play at random positions along the X axis (left and right of the screen). A random number is also used in the "Wait" command so that the timing of the repetition is also random.




There are two programs for the Queen. The first program sets the initial settings. As soon as the program starts, it turns off the display of Queen. The size should be increased by 2 in the X, Y, and Z directions, just like the knight and warrior. As for the coordinates, let's set the coordinates to X (22), Y (44), and Z (202) for her appearance with warp effect later.

When the "action" signal is received, play the effect "Level up" to express warp and play the sound (Magic 2). Then turn on the display of the queen character to make the queen appear.

So far, you have seen the programs of Example 1) to Example 4). There are many things to consider when creating a good-looking scene, such as object positions, angles, movements, camera settings, and how effects and sounds are synchronized with the footage. However, unlike the actual movie shooting, the same scene can be recreated many times by trial and error with slightly different conditions, so in the end it is very fun to create the scene you envisioned with Mind Render, and you can get the satisfaction of creating the work. By continuously executing the completed program, you can enjoy it as if you were playing a short movie. You can also use video editing software to add subtitles and create a work that people can see on YouTube or SNS. Mind Render comes with tons of characters, backgrounds, sounds and effects. Please challenge yourself to create works using your imagination.

# 13. Ways to learn more about Mind Render

Please refer to the Mind Render official website for new releases of Mind Render and the latest related information.

https://mindrender.jp/
Mind Render Official Website / Mobile Internet Technology Co., Ltd.

If you want to know how to use Mind Render, please find related videos on You Tube.

1) Official YouTube channel
https://www.youtube.com/channel/UCrTW9l55kjidhKkCpf7NZHg

2) Mind Render Information Channel
https://www.youtube.com/channel/UCx1ocvTHf_DxjlMTD30zthQ

For detailed programming explanations, please refer to the following books.

“Nurturing a Genius Programming Workbook on Mind Render (English Edition) ", by Ryoichi Shirato and others, published by Cut System (ISBN-10: 4877834362, ISBN-13: 978-4877834364)

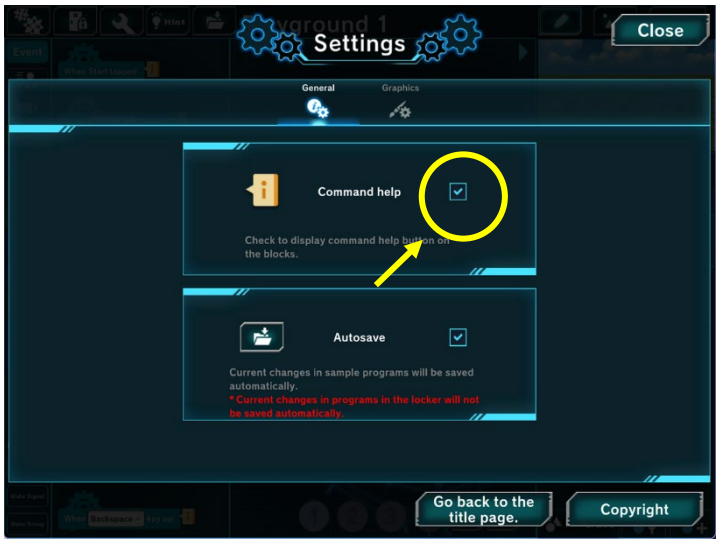
Amazon : https://www.amazon.co.jp/dp/4877834362/ref=cm_sw_em_r_mt_dp_-O0ZFbMB7TN0F

Note: This book is based on an early version of Mind Render. Since some of the screens and instructions in Mind Render 2 have changed, there are some differences from the explanation in the book.

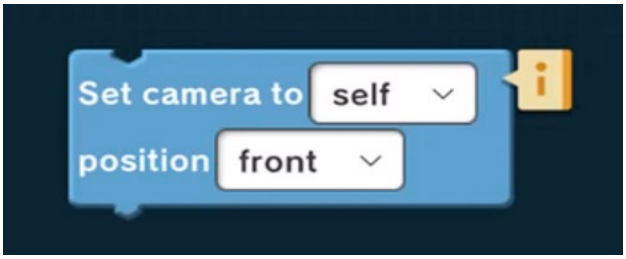
# 14. Tips for using Mind Render 2

## 1 ) How to see the command help

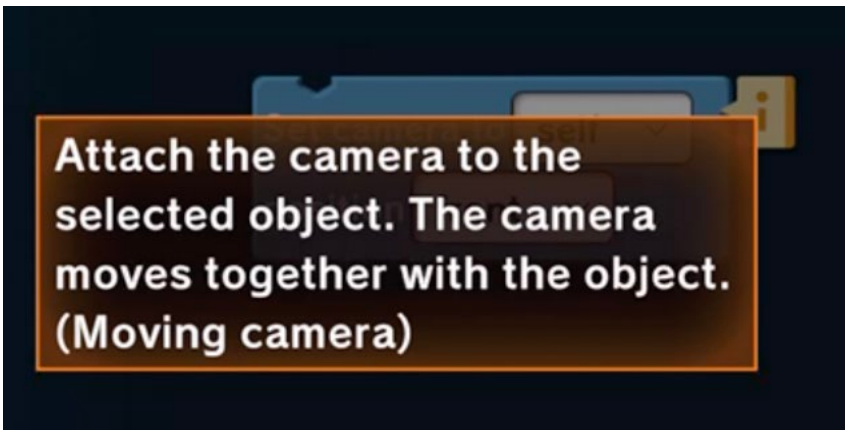
Open the Settings screen.




In the Settings screen, tap the checkbox next to “Show command help button”. When you are done, tap "Close" in the upper right corner of the screen. (Note: The screen resolution slider is only visible in the MacOS version.)




An "i" will appear on the right side of the command block. Tapping on this "i" will display an explanation of how to use this command as shown below.

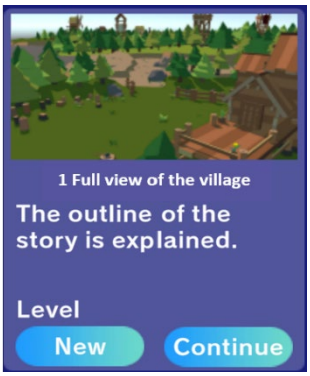


## 2）I want a clean display and smooth movement

If you move the "Screen resolution" slider on the “Graphics” tub on the setting screen to the right, it will be displayed clearly with a high resolution. Instead, the processing power of the computer is also required, so the movement may look awkward depending on the performance of the computer or tablet you are using. In such cases, moving the slider to the left will smooth out the movement. (Note: This feature is only available on MacOS version.)


## 3）I want to save the created program so that it does not disappear

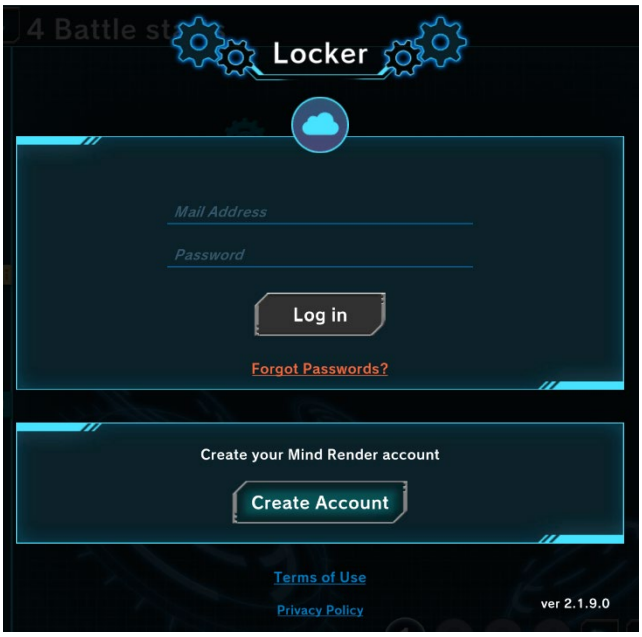
Mind Render keeps the created program in the last used state.



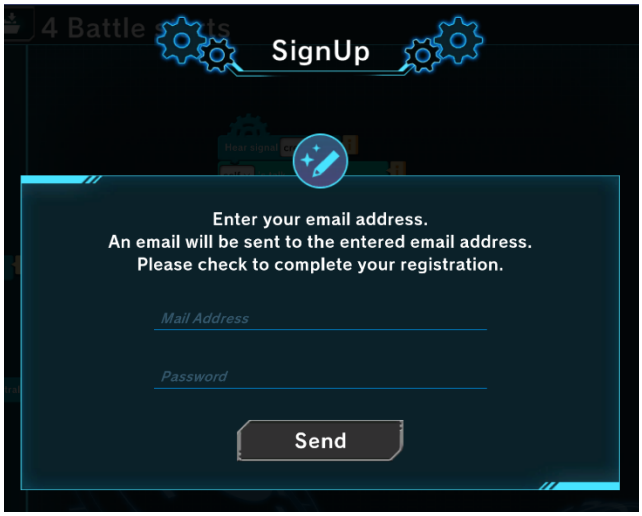

You can start Mind Render again from where you left last time by tapping the "Continue" button that appears when you click one of the labs in the lab menu.

When you click "New", what you made last time disappears. When creating a new program using the same lab, use a "Locker" if you want to keep the old program as well. Locker is a storage space on the cloud, where you store your programs. We will explain how to use the locker.

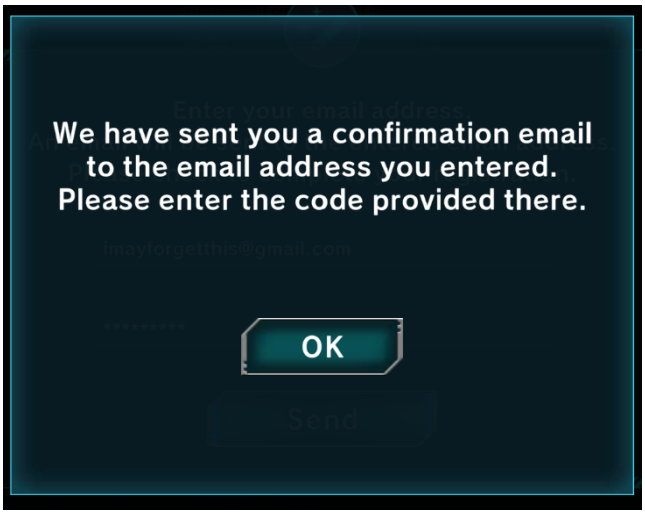
Click the button in the middle of the menu button on the top left of the screen.



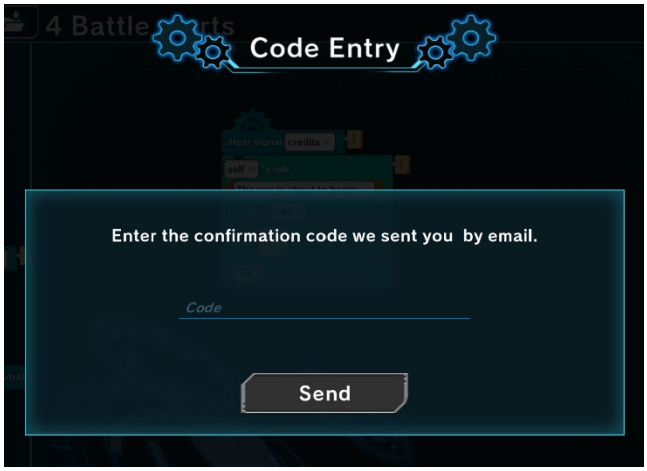

The locker screen will appear. When you use it for the first time, tap on the “Create Account" button.




Enter your email address, set your password, and click the "Send" button.

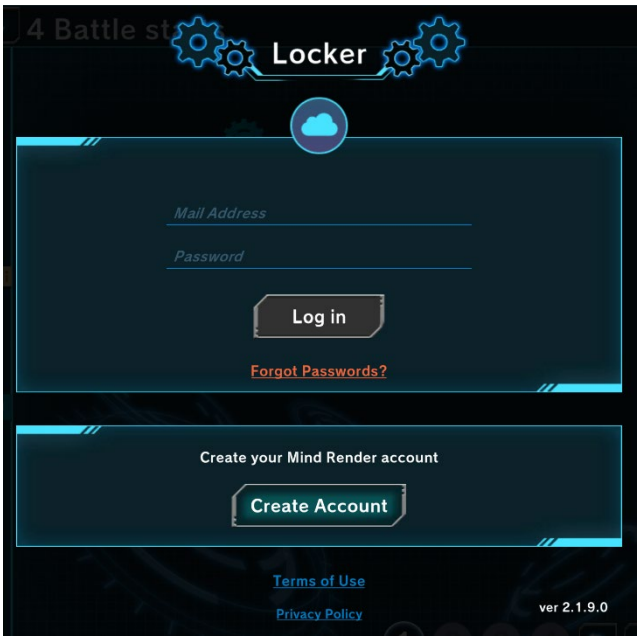



Click the "OK" button.



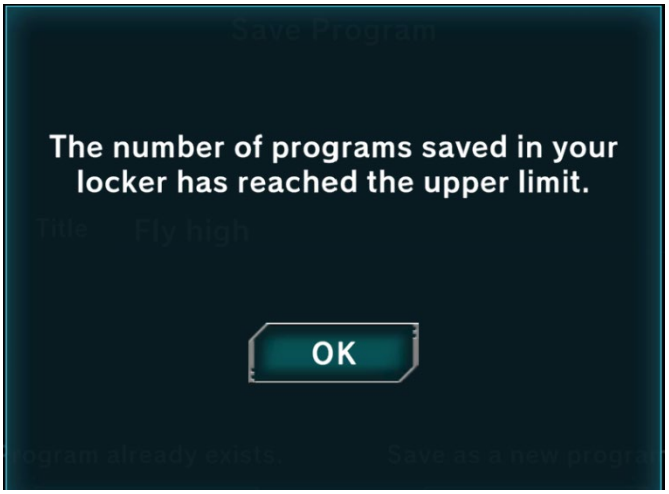

The Code Entry screen will be displayed.
A code entry screen is displayed.
A confirmation e-mail will be sent to the registered e-mail address, so enter the code described in the e-mail.
Click the "Send" button to complete the registration.

Keep your password so you won’t forget it.




Enter your registered email address and password and click the "Log in" button.


When you log in to the locker, the screen on the left will be displayed. To save the current Mind Render program as a new one, click "Save As". You can freely name the program when saving a new program in the Title line. If the program has already been saved in the locker and you want to overwrite it, click the "Overwrite" button.



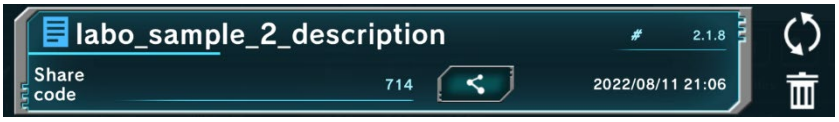

Note: Up to 10 programs can be stored in the locker. If you try to save more than 10 programs, the screen below will appear.


You can retrieve the saved program and enjoy it or click "My Locker" to check the contents of your locker when you want to continue. By clicking the "Load" button on the right of the program you want to retrieve, you can load the program on your computer or tablet.

Click the "Delete" button to remove programs that are no longer needed. When you have finished using the locker, click the "Close" button in the upper right.

## 4）I want to share the program I made with my friends

You can share the programs saved in My Locker with your friends.

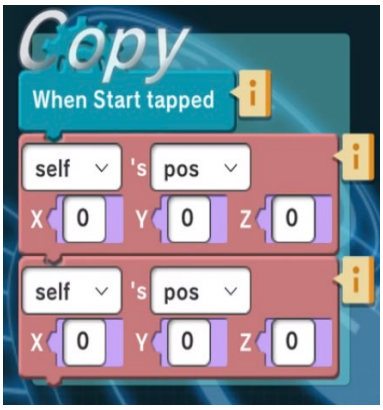



Set a "Share code" for the program you want to share.

A "Share code" is something you give to someone you want to share your program with so that they can see and use your program.


Conversely, if you want to view or use a friend's program, ask for the shared key, and enter the shared code in the "Shared locker" field to load the friend's program.

## 5）Copy function




You can easily copy the created program.
Long press (or long tap) the left mouse button on the top block of the part you want to copy.
Copying is completed when the word "Copy" appears on the left shoulder of the block and disappears.

If you press and hold the top block, all the blocks after that will be copied, and if you press and hold a block in the middle, the block and below ones will be copied.
Press and hold the left mouse button (or tap and hold) on an empty space in the programming area to paste.

You can paste between different objects, even between different missions.

## 6）Adding objects

Objects can be added from the "Add Object" button at the bottom right of the object list


## 7）Changing object settings

You can set the position and size of the object, various settings of the physics engine, etc. programmatically, but you can also set it directly on the object detail screen.

In the object list, select the object you want to change and click the magnifying glass button at the top of the list.

Open the object details screen. You can directly set the position and size of the object, various settings of the physics engine, etc.


## 8) Sample Programs

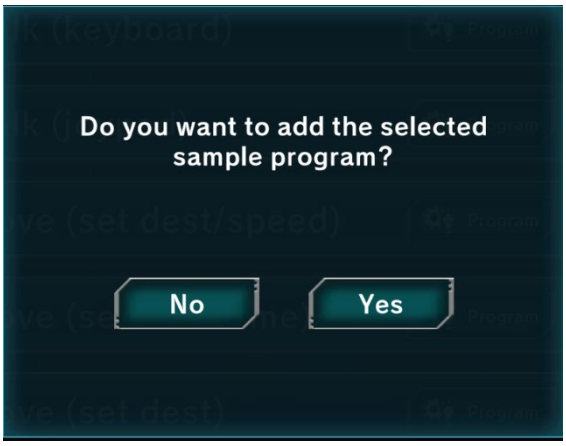
Sample programs and hint videos are very useful when you want to find out how to use commad blocks.

At the top left of the Mind Render program editor screen is a "Hint" button, tap it to open a menu of hints.

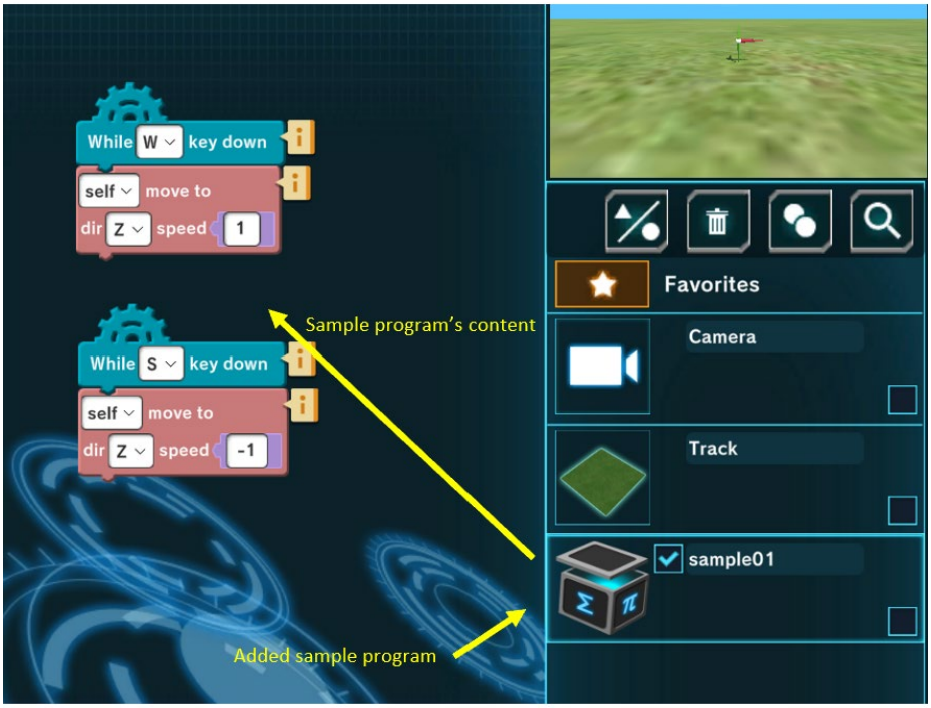

The figure below shows the menu of the Help Sample Program when opened. Many topics are displayed by scrolling, so you can search for commands and things you want to do by yourself.



Each topic has two buttons as shown on the left. (There are also topics with only hint videos, without sample programs.)

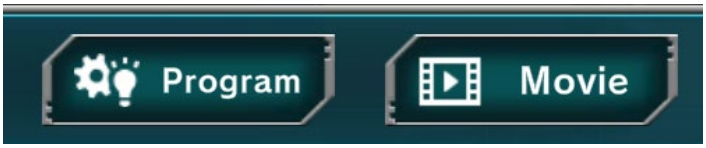



Click the "Program" button to open the screen shown on the left. Click Yes to load the sample program into the currently open lab (program).




A new object called "sample01" is added to the object list. Click this object to display the contents of the sample program.

Click the "Movie" button to see a hint video





* Subtitles are provided only in Japanese language.

## 9）Recording

You can use your computer or tablet's microphone to record your own voice and sound effects and use them within the program.

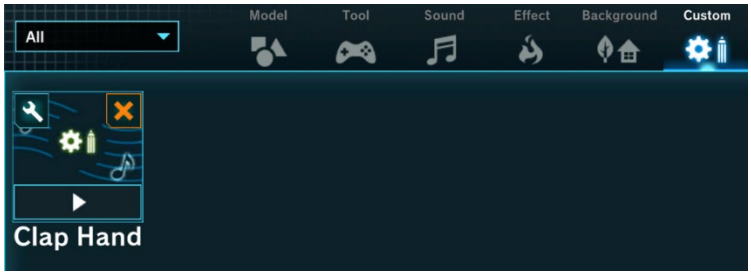
If you select Sound from the Object menu, you will see a button with a "+" sign on the upper left. Click this button.


A recording screen like the one on the left will open.

In the "Name" field, enter the name you want to retrieve later.

When you are ready to record, press the red circle button to start recording.

If it doesn't work, you can try again and again. When you are done recording, click the "Save" button to save.




Saved sounds can be found in the object menu "Custom" as shown on the left figure.


If you add it with the program editor, you can call it within the program。

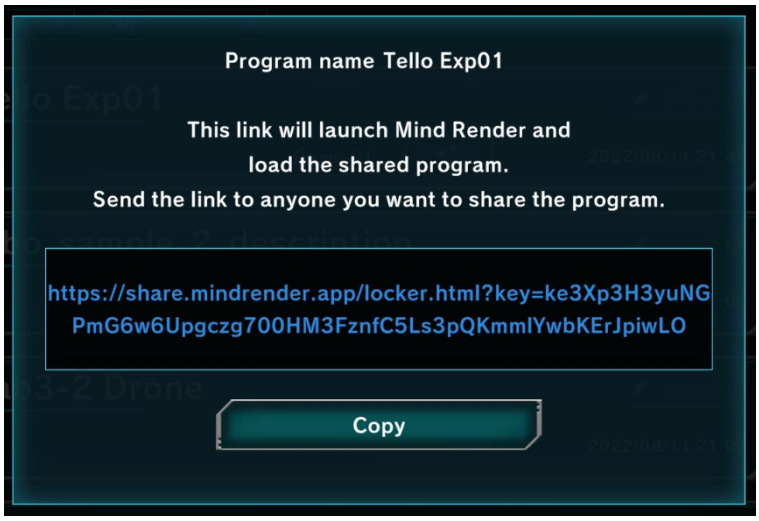
## 10）Sharing programs using shared links

By sending a link, you can easily share your program with anyone.

### 1）When you want to send someone your program

After saving the program you want to share in the locker, click the "My Locker" button in the locker to display the program list.

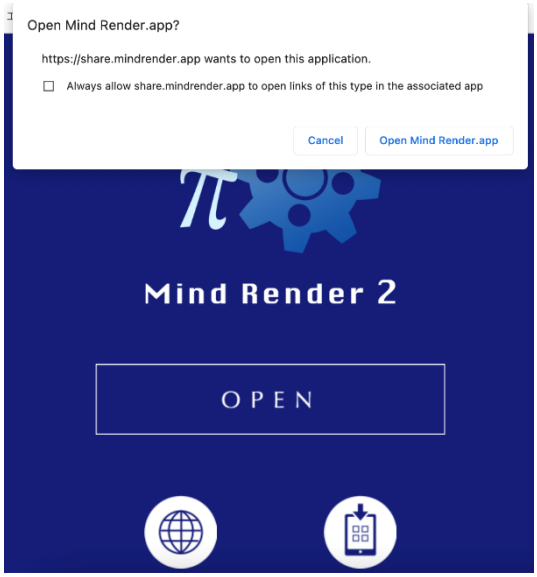

Find the program you want to share and click the "Share button" shown on the left figure。



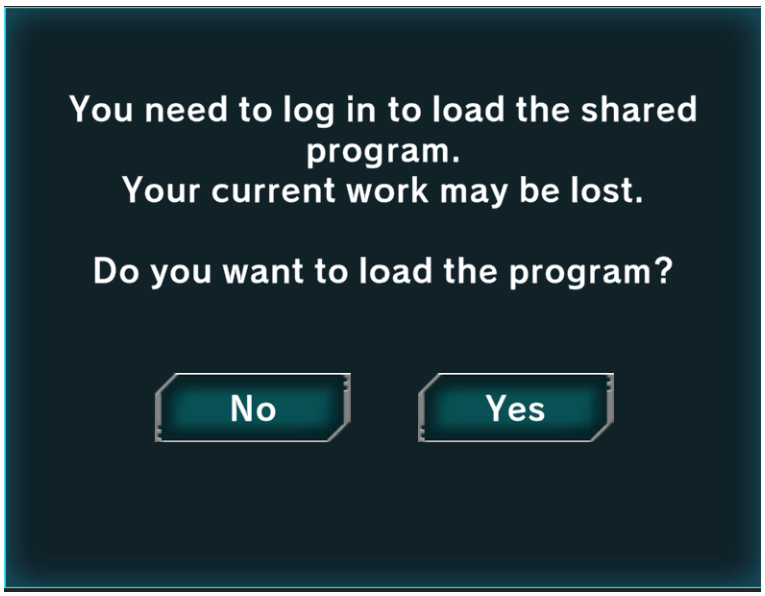

The screen shown on the left will open. Click "Copy" to copy the URL for sharing to the clipboard of your computer or tablet.

### 2）When someone sends you a program sharing link

If you receive a shared link by e-mail, etc., click the link and the screen below will be displayed.




Click “Open Mind Render.app”.

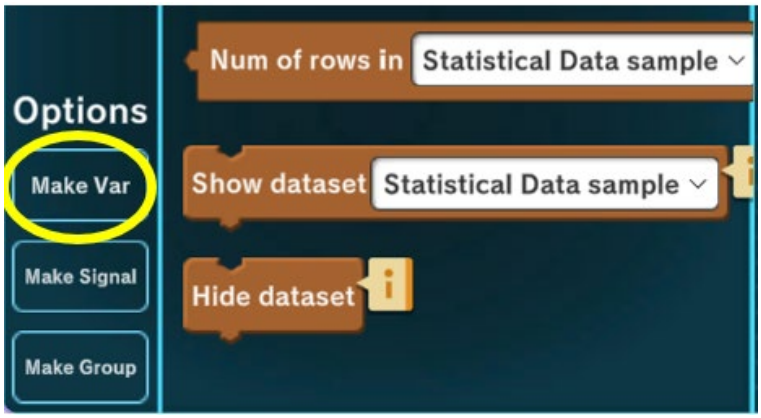



Mind Render will open as shown on the left figure, and a message to confirm loading is displayed.

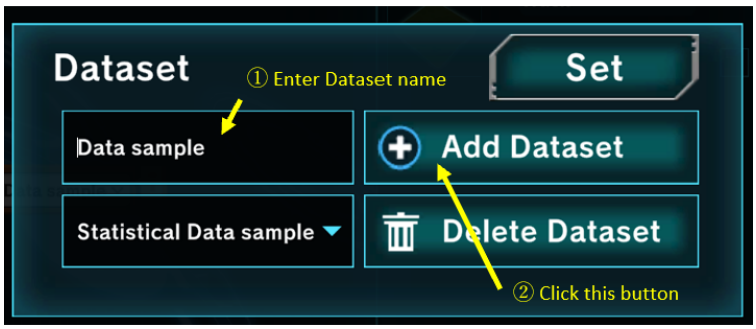
Click "Yes" to load the shared program.

## 11) Statistical data

You now have access to statistical data for use in graphs, etc. You can load it into Mind Render by following the steps below.



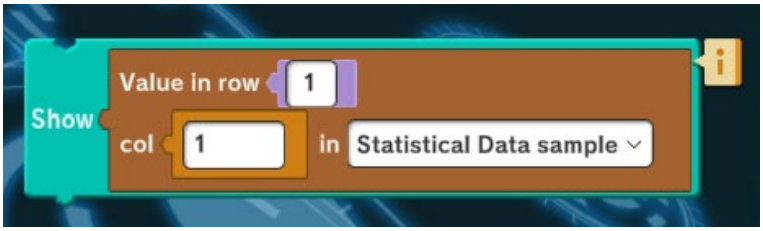

Click the "Make Var" button at the bottom left of the program editor screen.




Enter the dataset name in the lower right "Dataset" of the displayed "Make Var" screen and click the "Add Dataset" button on the right side. Press the "Set" button.


The "External data" screen shown on the left figure is displayed. Check the contents and click the "Download" button.


When loading is completed, the data will be displayed in the area on the right as shown on the left figure.




The loaded dataset can be called and used within the program.

A quick start guide to Mind Render programming
Published on August 22, 2022, ver1.6

Author: Mobile Internet Technology Co., Ltd.
Published by Mobile Internet Technology Co., Ltd.
〒 Kioicho Park Building 1F, 3-6 Kioicho, Chiyoda Ward, Tokyo 102-0094, Japan