Australian **Computing** Academy

DT Challenge Blockly

Space Invaders

1. The journey begins

2. Growing the fleet

3. Stardrive operation

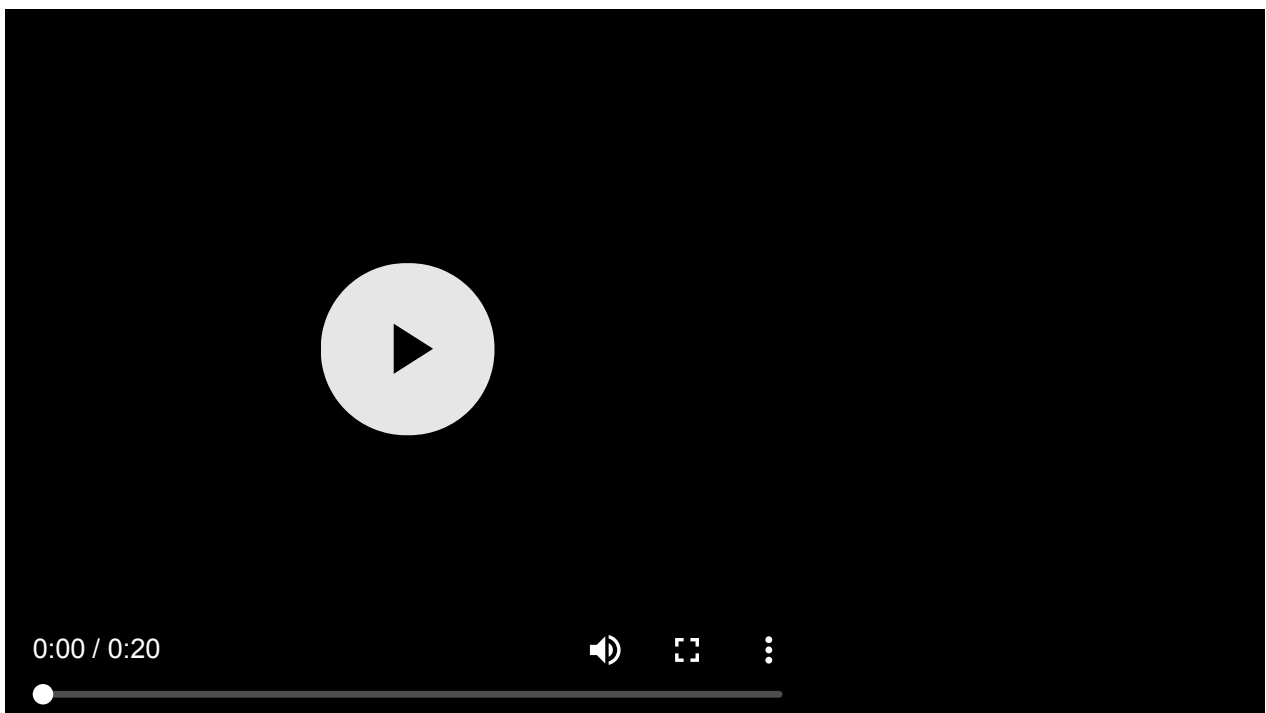4. The battle for Earth

1

THE JOURNEY BEGINS

1.1. The journey begins

1.1.1. Planet Zarg

Planet Zarg is suffering from a terrible drought. Most of the water has evaporated into space.

You are the Commander of the alien fleet from planet Zarg.

In this time of crisis, the people of Zarg need your help.


0:00 / 0:20

Planet Zarg (https://groklearning-cdn.com/modules/Y6xAC38rZ6JenitbkLfrmd/planet_zarg480p.mov).

1.1.2. Mission briefing

The only other known source of water in the universe is on planet Earth.

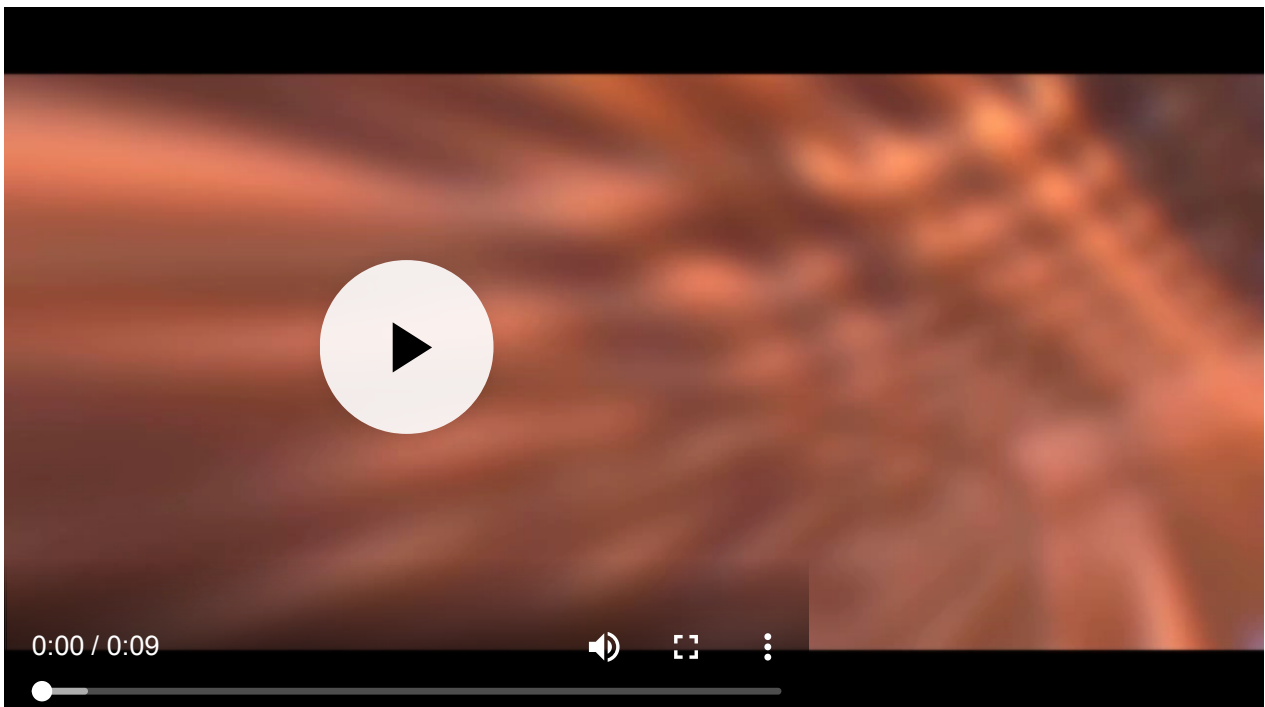Your mission is to build a fleet of spaceships to get hold of Earth's abundant water supplies.

Be advised, the United Nations of Earth has a powerful defence system.

Earth (https://groklearning-cdn.com/modules/Z2fp8ziBcAYrLuedYEomdN/securewater480p.mov).
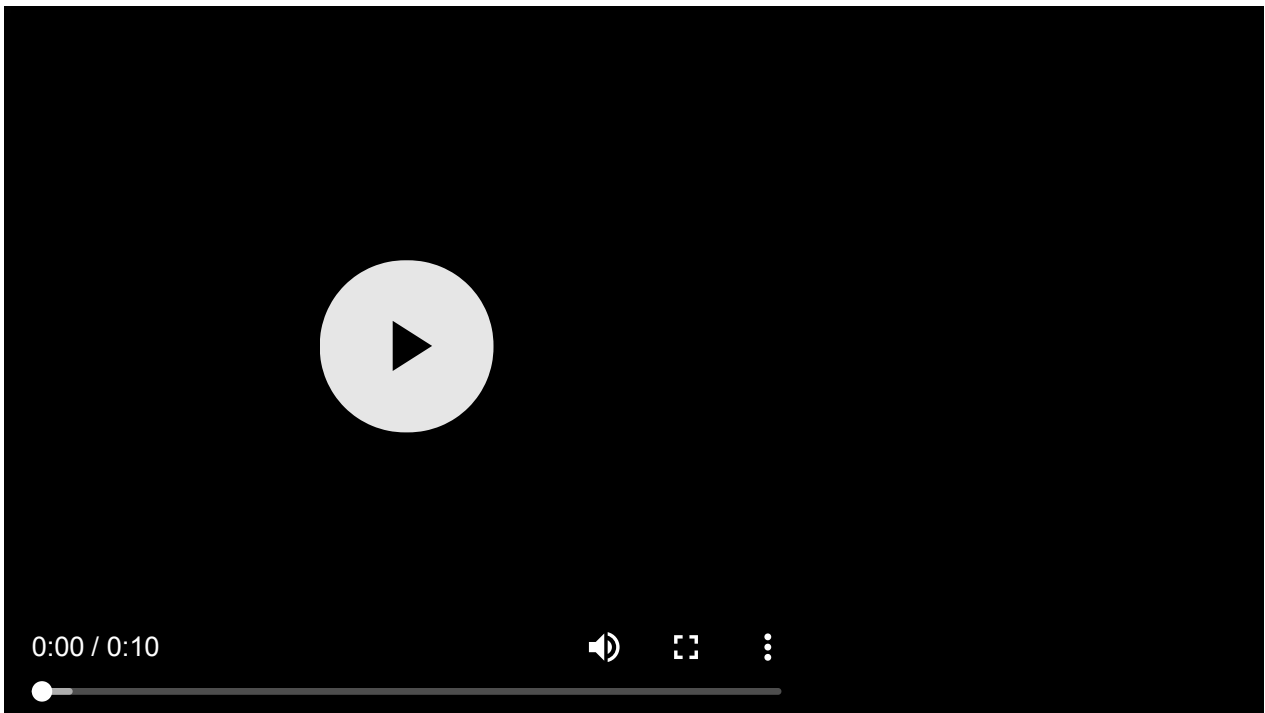
1.1.3. Space station

Commander, for the past five years, Zarg's best engineers have constructed a space station in orbit. There, we are building the trans-warp spaceship that will take you to Earth safely.

Zargan Space Station (https://groklearning-cdn.com/modules/2XsrgWUDp73rvLGofzC3MG/zargan-space-station.mov).

1.1.4. Good luck and Godspeed

Commander, Mission Control has informed us that our spaceship is ready for launch. Engaging impulse drive to leave orbit.
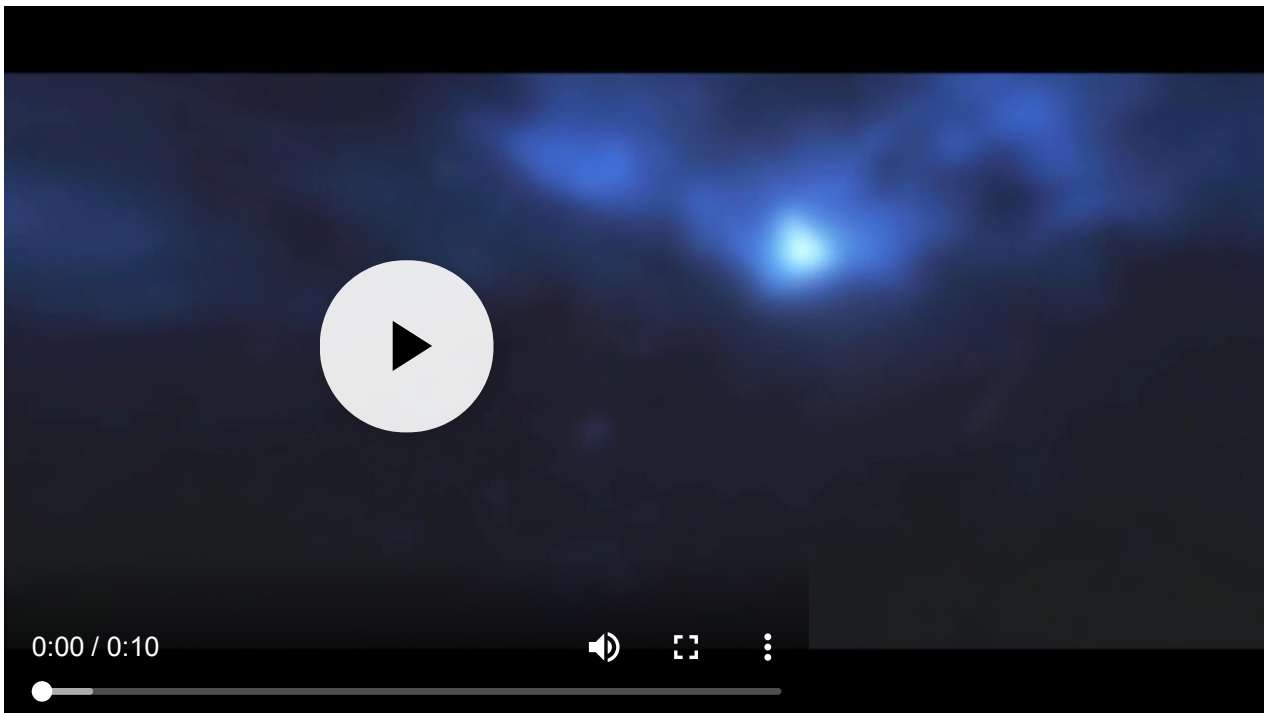
We are 'go' for Launch (https://groklearning-cdn.com/modules/BaNrDDHE8wz8jiAe9EfrCn/spaceship_leaving_zarg.mov).

1.1.5. The end goal

Commander, we are building the fleet on our long journey to Earth. Our science department is still working on some of the details, so you will be given access to top-secret command protocols as they become available.

Your end goal is to build a fleet and get some of Earth's water resources.

(https://groklearning-cdn.com/modules/kPDtAbVNRtj9kuCSoTdwa8/endgoal.png)

The Zargan Fleet

We are receiving a transmission from the Zargan Science Department ... first set of command protocols available ... upload link activated ... uploading ... upload complete ... you are cleared to begin building your fleet. Good luck, Commander.

1.1.6. Handy hints

Commander, on the **next page**, you will start coding. Here are a few useful hints up front.

💡 How do I write and check my code?

1. Write your program in the editor
   (large panel on the right);

2. Observe the output in the Preview window. Check that the program works correctly and compare its output with the screenshot!

3. Mark your program by clicking ⭐ Mark and we will automatically check if your program is correct, and if not, give you some hints to fix it up.

1.1.7. Problem: Making our first spaceship ⌨

Commander, your first task is to **make one spaceship appear in space.**

You have received access to top-secret command protocols.

Use the computer command interface on the right.

Follow the steps

Step 1: Prepare the simulator

All our code goes into this block.

Step 2: Create an empty spaceship

Step 3: Choose an image for the spaceship

Put the code together and this is the result:
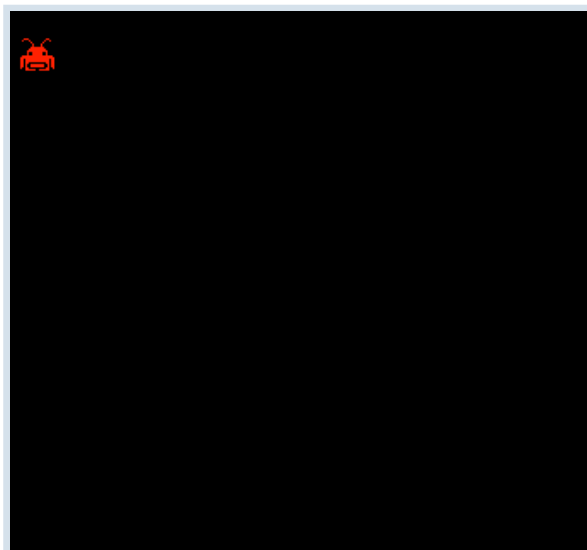
 (https://groklearning-cdn.com/problems/zBAo5pwkE5TBtmcCe8XiYX/si-step0.png)
Click to open full-size.

Congratulations, Commander. We have taken the first step towards the invasion of planet Earth.

You'll need

📄 program.blockly

Testing

☐ Testing your first spaceship. Good job, Commander. Your code has passed our test.

1.1.8. Problem: Changing the appearance of a spaceship

Commander, the science department has developed a new spaceship design that will make it impossible for Earth's defences to lock on. They want you to test it.

Change Spaceship Design

To change the spaceship design, change the block with the spaceship image 👾 . Look at the editor on the right where you see the code from the previous mission.

new spaceship with image 👾

Change the spaceship image until you see this spaceship:

 (https://groklearning-cdn.com/problems/Uq8eUrHKDx73VTCW8puDR2/spaceship3-screenshot.png)

Spaceship2 Design

Congratulations, Commander. This is a cool-looking spaceship.

You'll need

📄 program.blockly

in space
do   set spaceship to   new spaceship with image 👾
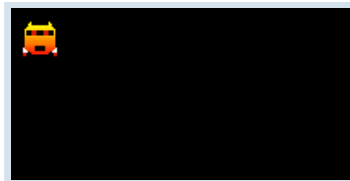
Testing

☐ Testing the design of the first spaceship. Good job, Commander. Your code has passed our test.

1.1.9. Problem: More spaceship designs

Commander, the science department has developed more cool-looking spaceship designs. You now have access to additional top-secret spaceship skins.

Choose one of these new looks:

 (https://groklearning-cdn.com/problems/6RLTS6u7saaKRRovWfpsSD/spaceship4-screenshot.png)

spaceship3 - Inferno
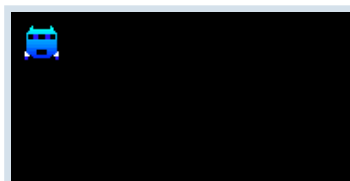
 (https://groklearning-cdn.com/problems/zqBhB5puN9GgF2irPVBaZ7/spaceship5-screenshot.png)

spaceship4 - Blizzard
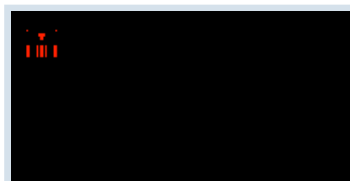
 (https://groklearning-cdn.com/problems/F3yq6LGLaa2y5odjDrFbLX/spaceship6-screenshot.png)

spaceship5 - Stealth
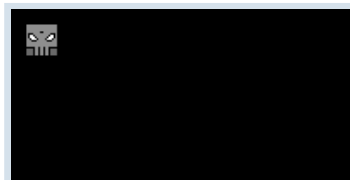
 (https://groklearning-cdn.com/problems/p3RDwkg4UdCKkiw2veW6MH/spaceship7-screenshot.png)

spaceship6 - Skull
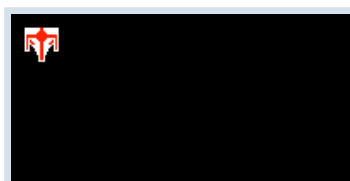
 (https://groklearning-cdn.com/problems/npp3dW7VDna5uxdkVZbkU2/spaceship8-screenshot.png)

spaceship7 - Xavier
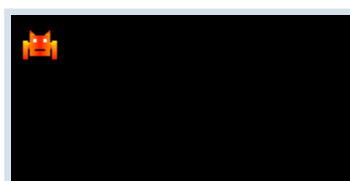
 (https://groklearning-cdn.com/problems/z5nVXyfYxCZk9rA27hmHDJ/spaceship9-screenshot.png)

spaceship8 - Wildfire

Congratulations, Commander. The Earthlings will be impressed.

In any of the following questions, you can use your favourite design, or change designs as you like. We will show you the solutions with spaceship1.

https://aca.edu.au/challenges.html

You'll need

📄 program.blockly
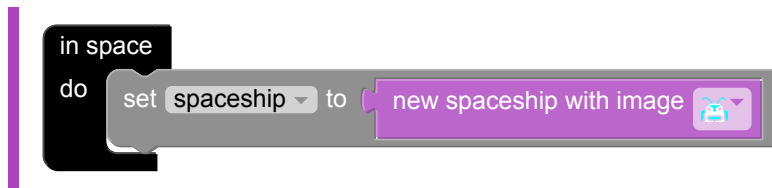
Testing

☐ Testing your first spaceship. Good job. Your code has passed our tests.

1.1.10. Slingshot around a black hole

Commander, the helm advises that we are about to slingshot around a black hole to pick up speed on our journey. As a safety precaution, we will put the ship on yellow alert. Put on your spacesuit and enjoy the view from the observation deck.

0:00 / 0:19
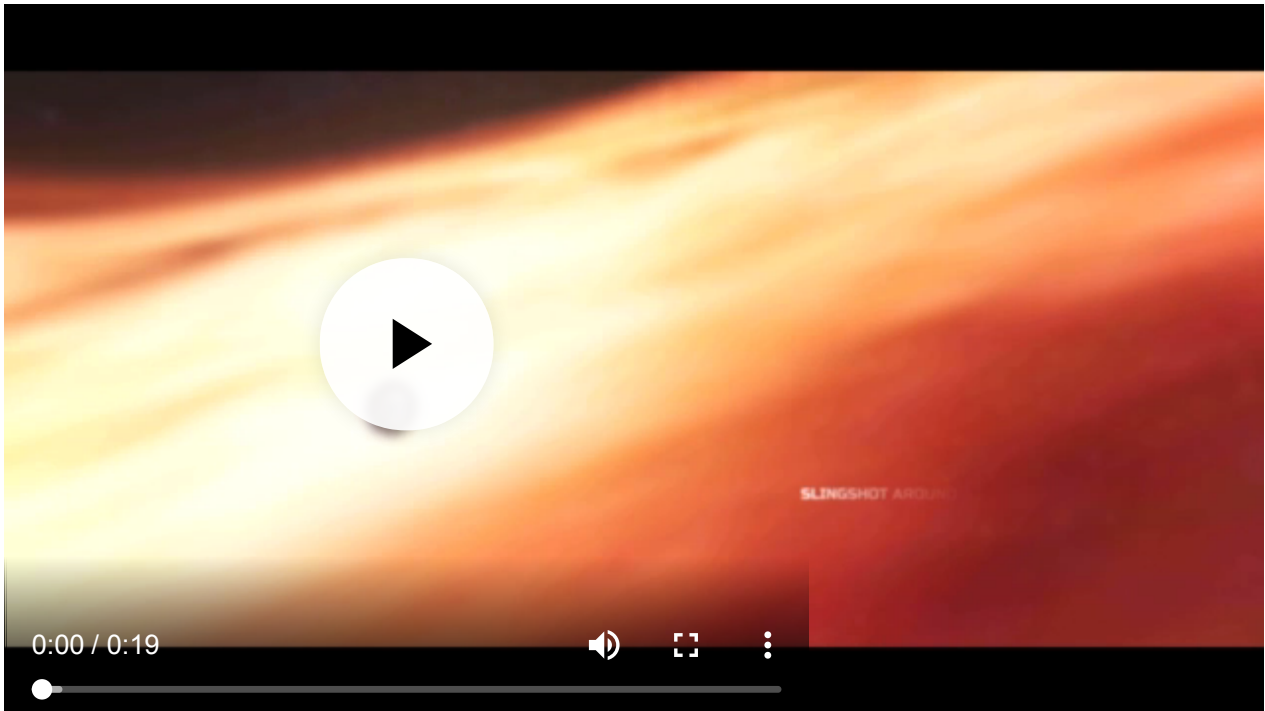
Black Hole (https://groklearning-cdn.com/modules/JEovCdw49Sd9BQPrhUXBzb/black_hole.mov).

1.1.11. Positioning spaceships

Commander, let's work out how to position spaceships. Our maths geniuses have found an elegant way to do this.

They say that we can set the distance from the top and the distance from the left for each spaceship.

In the video and image below, you see a single spaceship, which is positioned 200 pixels from the left and 100 pixels from the top.
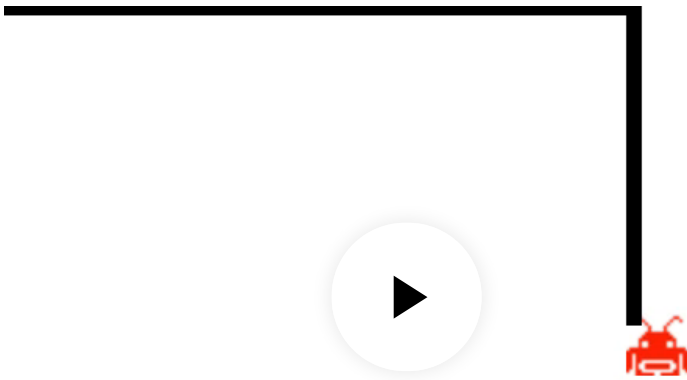
0:00 / 0:08

distance from top: 100px
distance from left: 200px

Position of a Spaceship (https://groklearning-cdn.com/modules/rTFcfuSyGe6zeEeBHFD69X/cartesiancoordinates.mov).
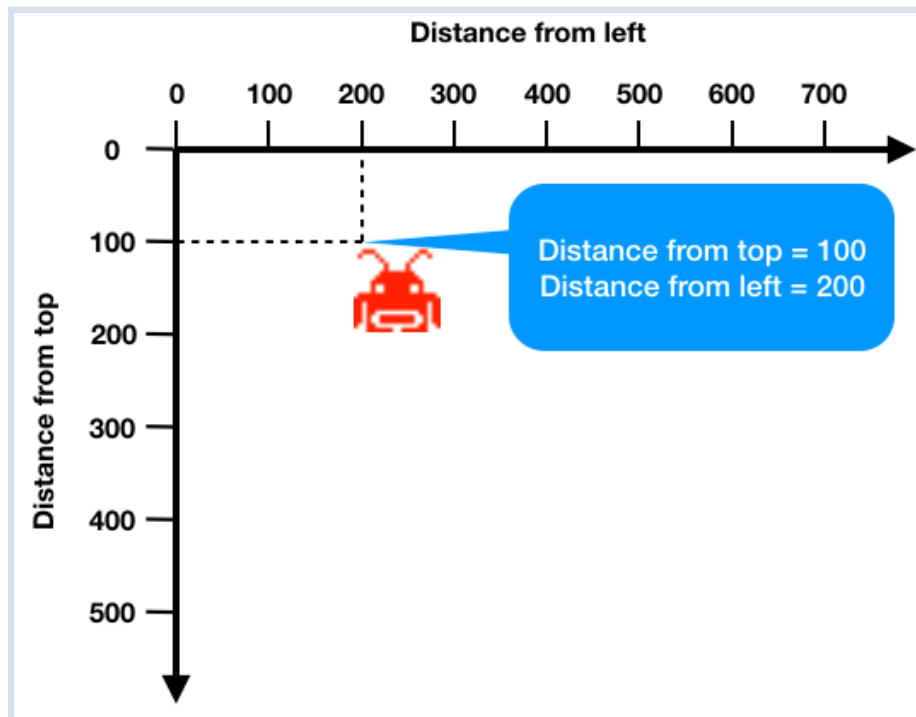
(https://groklearning-

cdn.com/modules/DhWwLw58xjNweyYWm7jHfb/cartesiancoordinatesystem.png)

Position of a Spaceship

A pixel is a unit of space. Remember that this is top-secret information.

1.1.12. Problem: Positioning a spaceship ⌨

Commander, you have access to top-secret protocols to position your spaceships. We start with one spaceship.

Follow the steps

Step 1: Left position

To set the spaceship's position from the left, add a 'left position' block from the purple Space Invaders container. Then you need to:

- add a `spaceship` variable from the grey variables container
- add a position value from the `blue` numbers container
- change the position value from 0 to 200
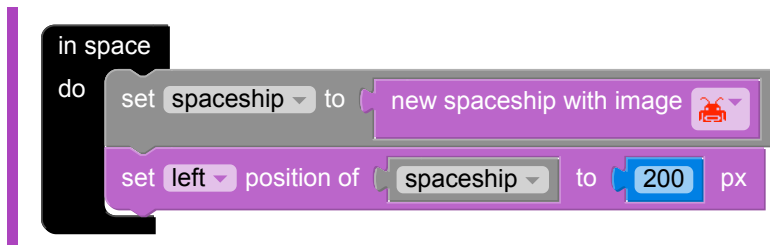
Step 2: Top position

The result should look something like this:
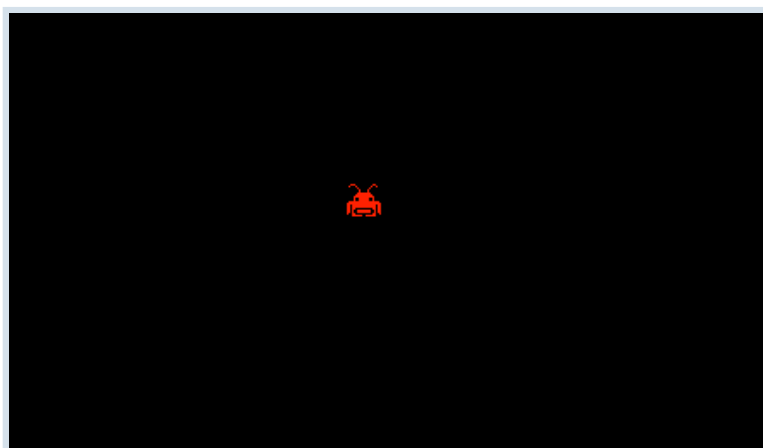
(https://groklearning-

cdn.com/problems/KEzcB9yznnJ3vDKPFrtrSK/si-step1.png)

A spaceship

Congratulations, Commander. You can position your spaceships freely.

We will practice this in the following mission.

You'll need

⟨⟩ program.blockly

```
in space
do    set spaceship ▾ to    new spaceship with image 👾▾
```

Testing

☐ Testing the position of the first spaceship. Good job. Your code has passed our tests.

1.1.13. Problem: Positioning a spaceship: manoeuvre ⌨

Commander, use your learning from the previous mission to position your spaceship:

- **277** pixels from the left
- **123** pixels from the top

Psst ...

Conduct this manoeuvre quietly so that Earth is not alerted to our presence. Our 3D Spaceship printer sometimes makes the space-time continuum vibrate. Earth now has super-sensitive detectors that can pick up gravitational waves.

You'll need

◈ **program.blockly**
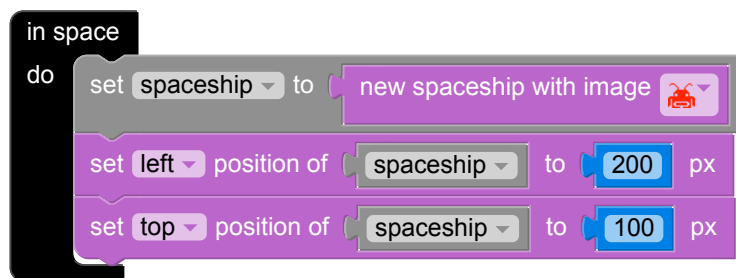
Testing

☐ Testing the position of the first spaceship. Good job. Your code has passed our tests.

1.1.14. Problem: Positioning a spaceship quiz

What are the correct commands to position a spaceship at 10 pixels from the left and 50 pixels from the top ?

○
```
in space
do    set spaceship to    new spaceship with image 👾
      set left position of spaceship to 50 px
      set top position of spaceship to 10 px
```

○
```
in space
do    set spaceship to    new spaceship with image 👾
      set left position of spaceship to 50 px
```

○
```
in space
do    set spaceship to    new spaceship with image 👾
      set left position of spaceship to 10 px
      set top position of spaceship to 50 px
```

○
```
in space
do    set spaceship to    new spaceship with image 👾
      set left position of spaceship to 277 px
      set top position of spaceship to 123 px
```

https://aca.edu.au/challenges.html

Testing

☐ That's right!

1.1.15. Captain

Commander, the Zargan High Command is following your efforts with great interest.

The High Command is **very** pleased with your progress of building the fleet and has promoted you to the rank of:

Captain

The Science Department, the Mathematics Department and all your Space Invaders congratulate you.

Long live the Zargan Empire.

Captain's Badge

2

GROWING THE FLEET

2.1. Growing the fleet

2.1.1. Mission briefing

Captain, you have successfully created a cool-looking spaceship. You know how to position it in space. The Zarg High Command has given you the authorisation to proceed with the fleet building program.

Your mission is to build a second spaceship and prepare to make many more.

Remember to conduct your activities quietly and in stealth so that Earth is not alerted to our activities.
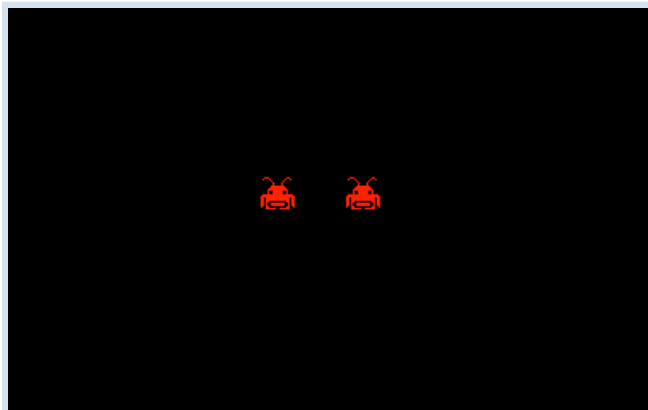
 (https://groklearning-cdn.com/modules/373gJwL7YMZPEszktefWsc/si-step2.png)
Two Spaceships

2.1.2. Problem: Making two spaceships ⌨

Captain, you are now going to build two spaceships.

- Your first spaceship shall be located at
  - 150 pixels from the left
  - 100 pixels from the top
- Your second spaceship shall be located at
  - 200 pixels from the left
  - 100 pixels from the top

💡 Building two Spaceships

- You will need a spaceship2 object. You can find it in the 'Variables' tray.

The result should look something like this:
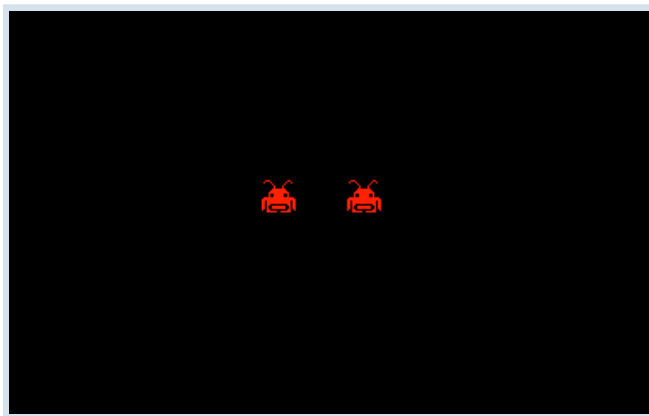
(https://groklearning-cdn.com/modules/373gJwL7YMZPEszktefWsc/si-step2.png)

Two Spaceships

💡 Hint

A common mistake is to use the 'spaceship' variable in the lower two purple blocks, rather than 'spaceship2'. So you might accidentally overwrite the positions of 'spaceship'. Make sure you use 'spaceship2'.

Congratulations, Captain. You have successfully made two spaceships.

You'll need

<> **program.blockly**

```
in space
do    set  spaceship  to  ( new spaceship with image  👾 )
      set  left  position of  ( spaceship )  to  ( 150 )  px
      set  top  position of  ( spaceship )  to  ( 100 )  px
      set  spaceship2  to  (
      set  left  position of  (  to  (  px
      set  top  position of  (  to  (  px
```
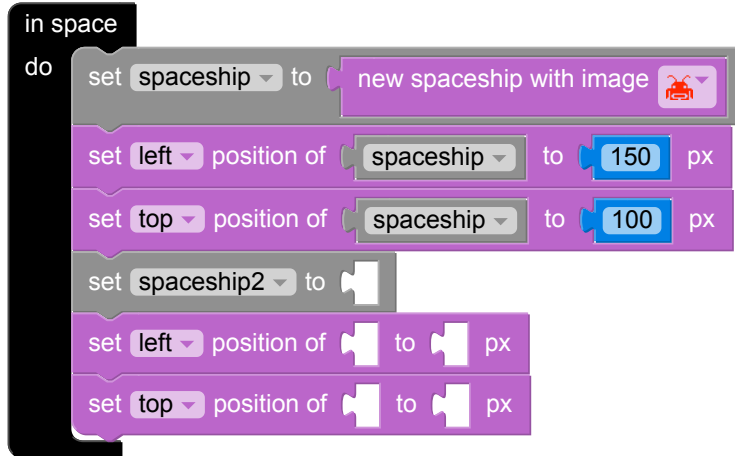
Testing

☐ Testing the position of two spaceships. Good job. Your code has passed our tests.

2.1.3. Emergency ...

Captain, the Science Department has advised us that we are running into serious problems.

They say that each time we create a spaceship, we need to add 8 additional blocks:
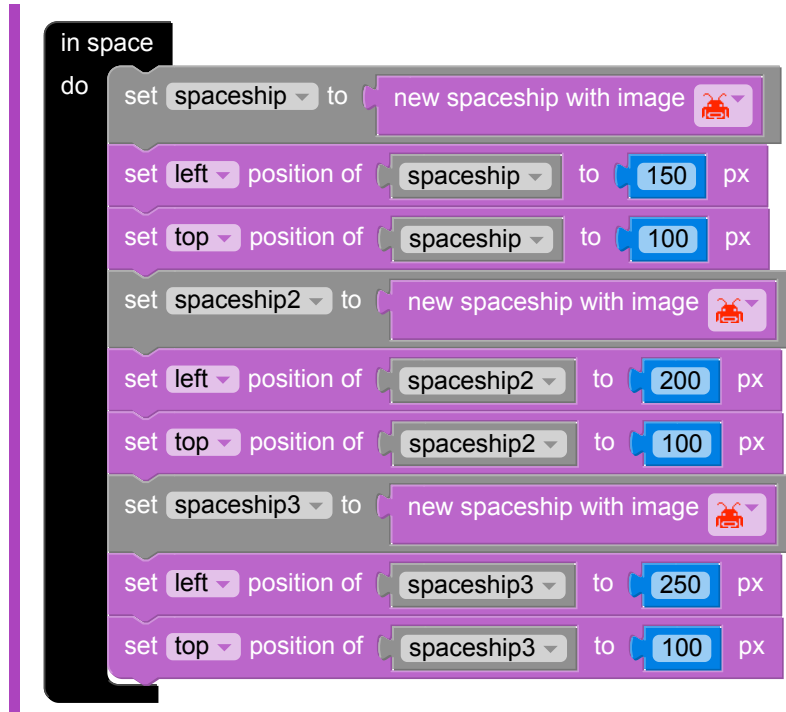
They fear that making spaceships this way will slow down the invasion of Earth. For 100 spaceships, we would need 100x8=800 blocks. The Science Department has done some deep thinking to make this more efficient and they have developed a new **function** toolset.

Commander, we have just received confirmation that the function toolset is now available for your immediate use.

2.1.4. Problem: Function toolset available

Captain, the Function toolset is at your disposal.

To use it, place the code that makes one spaceship into a function, `to create spaceship`

Then, you can call the function with `create spaceship with`. So, to make three spaceships, you call the function three times. Let's make three spaceships in a row, `100` pixels from the top and `150`, `200`, and `250` pixels from the left.

Put these three blocks **after** the `to create spaceship` block.

The result should look something like this:

 (https://groklearning-

cdn.com/problems/2f5ao4AKeXcorSGtTZQepZ/si-step3a-3spaceships.png)
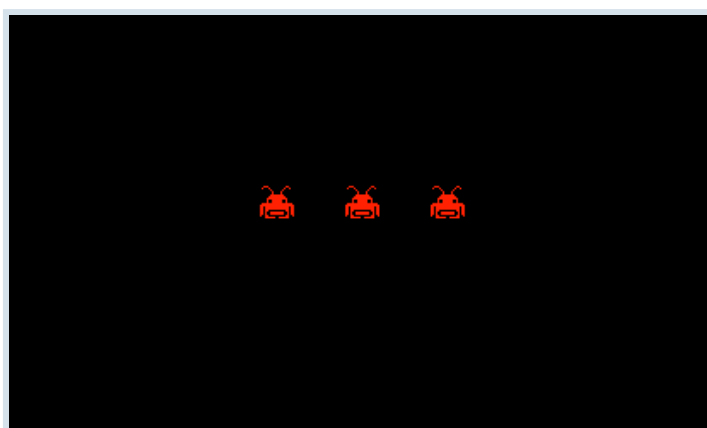
Three Spaceships

Captain, now we only need 1 line of additional code to make 1 additional spaceship. For 100 spaceships, that's only 100 lines of code, rather than 800. The invasion is on track again.

> 💡 **Hint - Click here to see the complete code**

You'll need

⬚ **program.blockly**

```
in space
do    to create spaceship with left , top
```

```
set spaceship ▾ to ( new spaceship with image 👾 ▾ )
set left ▾ position of ( spaceship ▾ ) to ( left ▾ ) px
set top ▾ position of ( spaceship ▾ ) to ( top ▾ ) px
```
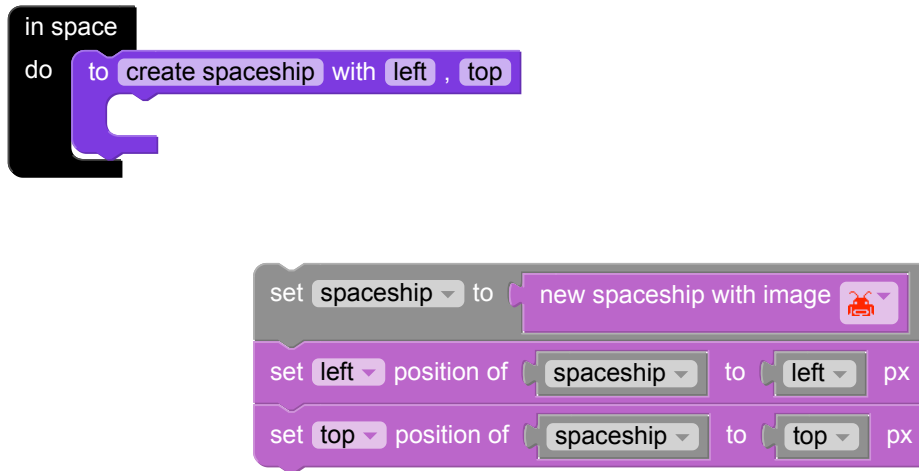
Testing

⬚ Testing the position of three spaceships. Good job. Your code as passed our test.

2.1.5. Problem: Row of spaceships: part 1

Captain, you are now building a row of spaceships.

- There should be 10 spaceships.
- They are located `50` pixels from the top.
- The leftmost spaceship starts at `30` pixels from the left.
- Each spaceship is 20 pixels wide and there is a gap of 10 pixels between two neighboring spaceships.

The result should look something like this:

 (https://groklearning-cdn.com/problems/pAiKyyAJ5VC4aZR3pqELBR/si-step4.png)

A Row of Spaceships

💡 Hint - Click if you are stuck

You'll need

📄 program.blockly

Testing

☐ Testing the position of ten spaceships. Good Job!

2.1.6. The Doctor

Captain, we will need to make several rows of spaceships for our end goal. The Science Department says that they think we can speed this up by making another function and a new thing they call a **loop** .

The chief scientist, only known as *The Doctor*, has recently received the Zargan Innovator of the Year award for this invention.

The Doctor offered to guide us through the whole process. Shall we give it a go?

2.1.7. Problem: Row of spaceships: part 2   ⌨

Captain, *The Doctor* is connected via hyperspace terminal from planet Zarg to support you with your first loop.

>hello, i am *the doctor* and am ready to assist you, captain.

>**Add** the following code under the `to create spaceship` block already in the window on the right.

```
repeat  10  times
do  create spaceship with  left  ◖ startLeft ▾   and  top  ◖ 50
    increase ▾  startLeft ▾  by  ◖ 30
```
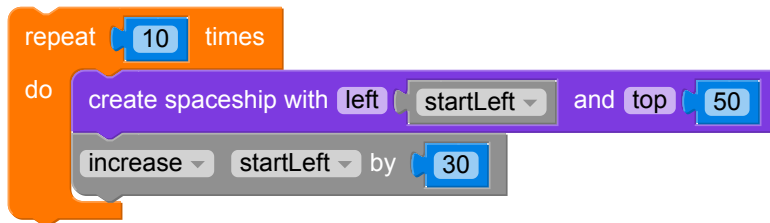
>the main idea is that we call `create spaceship with` function `10` times, but each time with a new value for `startLeft`.

>each time the loop runs, `startLeft` grows by `30` pixels. That's 20 pixels for the width of the spaceship, and 10 pixels for the gap between spaceships.

The result should look something like this:
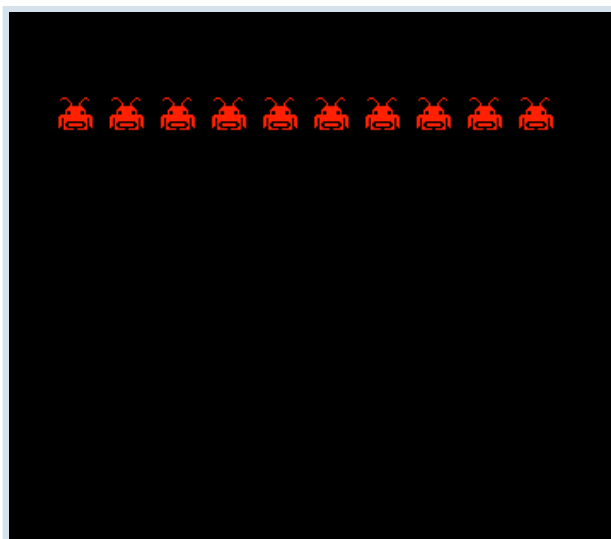
 (https://groklearning-cdn.com/problems/pAiKyyAJ5VC4aZR3pqELBR/si-step4.png)

A Row of Spaceships

💡 **Hint - Click here to see the complete code**

You'll need

program.blockly

```
in space
do
    set startLeft to 30
    to create spaceship with left , top
        set spaceship to new spaceship with image [🛸]
        set left position of spaceship to left px
        set top position of spaceship to top px
```
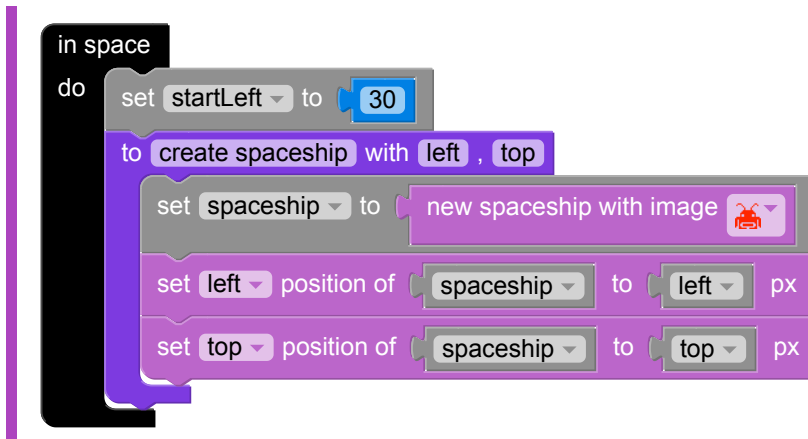
Testing

☐ Testing the position of ten spaceships. Good job.

2.1.8. Problem: Two rows of spaceships ⌨

Captain, well done on creating a single row of spaceships. Let's now make a second row.

- The second row is to be placed under the first row
- Make a gap of 10 pixels between two spaceships horizontally and vertically
- Each row starts at 30 px from the left.

♀ Hints

Use the code from the previous mission.

A spaceship is 20 pixels tall.

You will need a second loop.

Don't forget to set `startLeft` back to `30` before the second loop starts.

Here is a schematic drawing with the measurements you need.
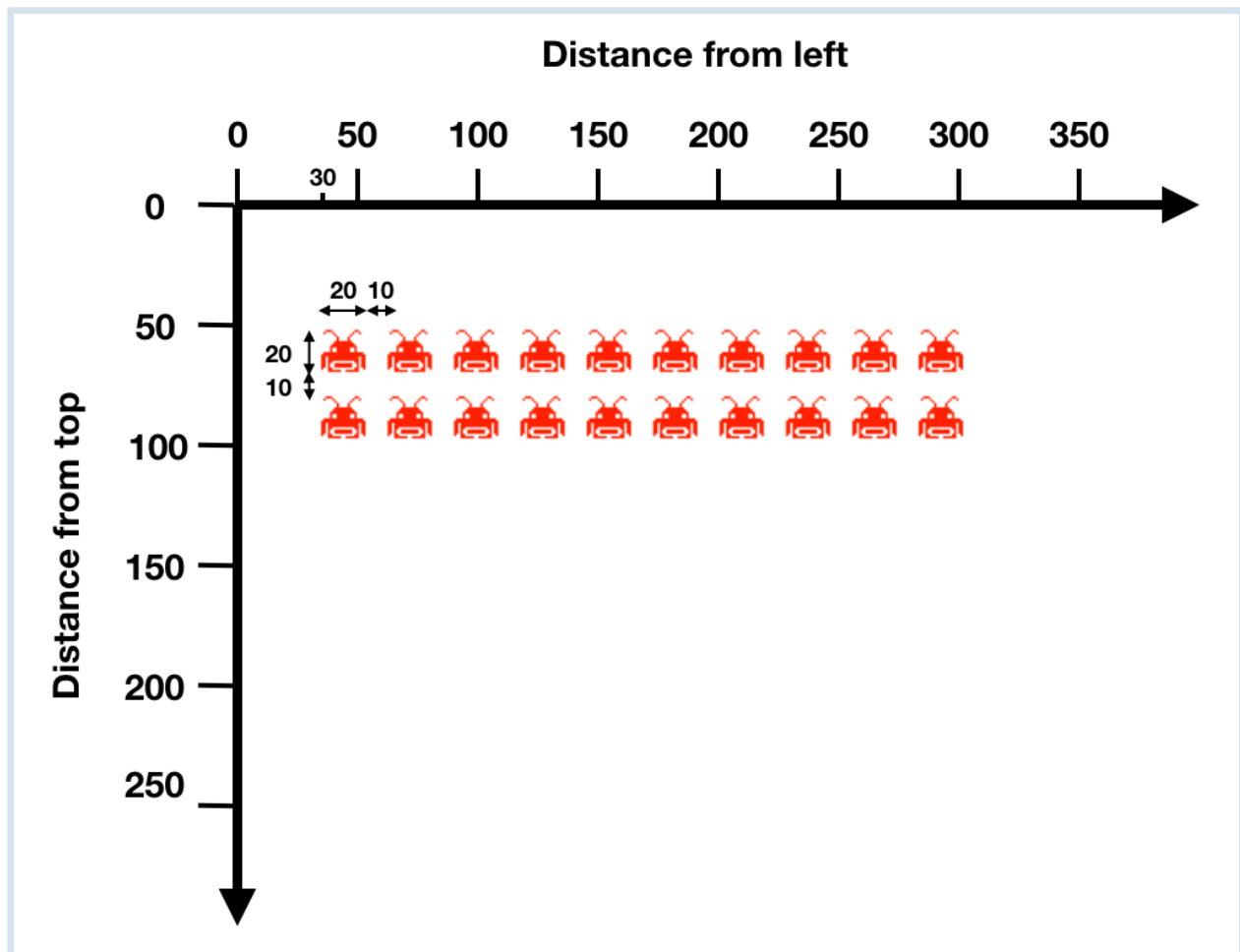
(https://groklearning-cdn.com/problems/nB8AnF6nAQ6RvkKKF77aFA/st-step4-22rowsofspaceships.png)

Two rows of Spaceships

The result should look something like this:

(https://groklearning-cdn.com/problems/CvqWdQhfwjMT3UajGebGYM/si-step4-2endresult.png)

Two rows of Spaceships

You'll need

📄 program.blockly
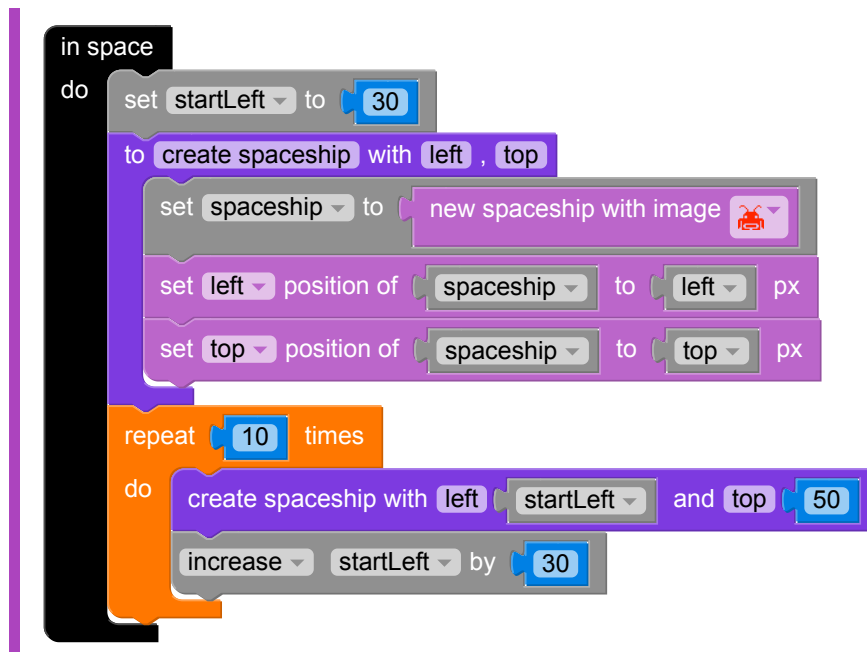
Testing

☐ Testing the position of two rows of spaceships. Good Job! Your code has passed our test.

2.1.9. Message from the science department

Captain, the Science Department has analysed your code.

They say that the repeat-loops only differ a tiny little bit by the distance from the top:
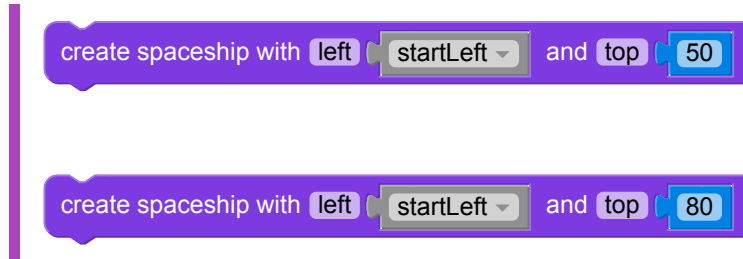
To make many rows of spaceships we can put that whole loop-code into a new function and call it as often as we need to.

This is similar to before when we made several spaceships.

Commander, the Science Department has uploaded new instructions. Let's go.

2.1.10. Problem: Three rows of spaceships ⌨

Captain, we now make three rows of spaceships.

Place the `repeat` **block into a new function** `to create row`

💡 Hints

We add a few more variables, such as the gap between spaceships, spaceship width and height and positions, so that we don't have to repeat their values in our code.

```
set gap ▾ to 10
set spaceshipsPerRow ▾ to 10
set startLeft ▾ to 30
set spaceshipWidth ▾ to 20
```
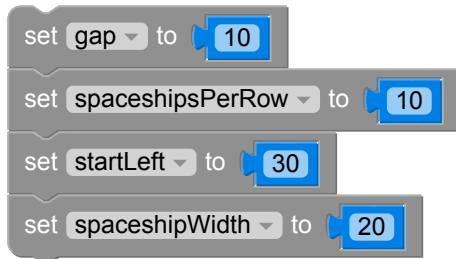
We can then use `create row with` as often as we need, for example three times to make three rows of spaceships.

- The first row starts `50` pixels from the top
- All rows start at `30` pixels from the left
- There is a gap of `10` pixels between each spaceship above, below and to the side
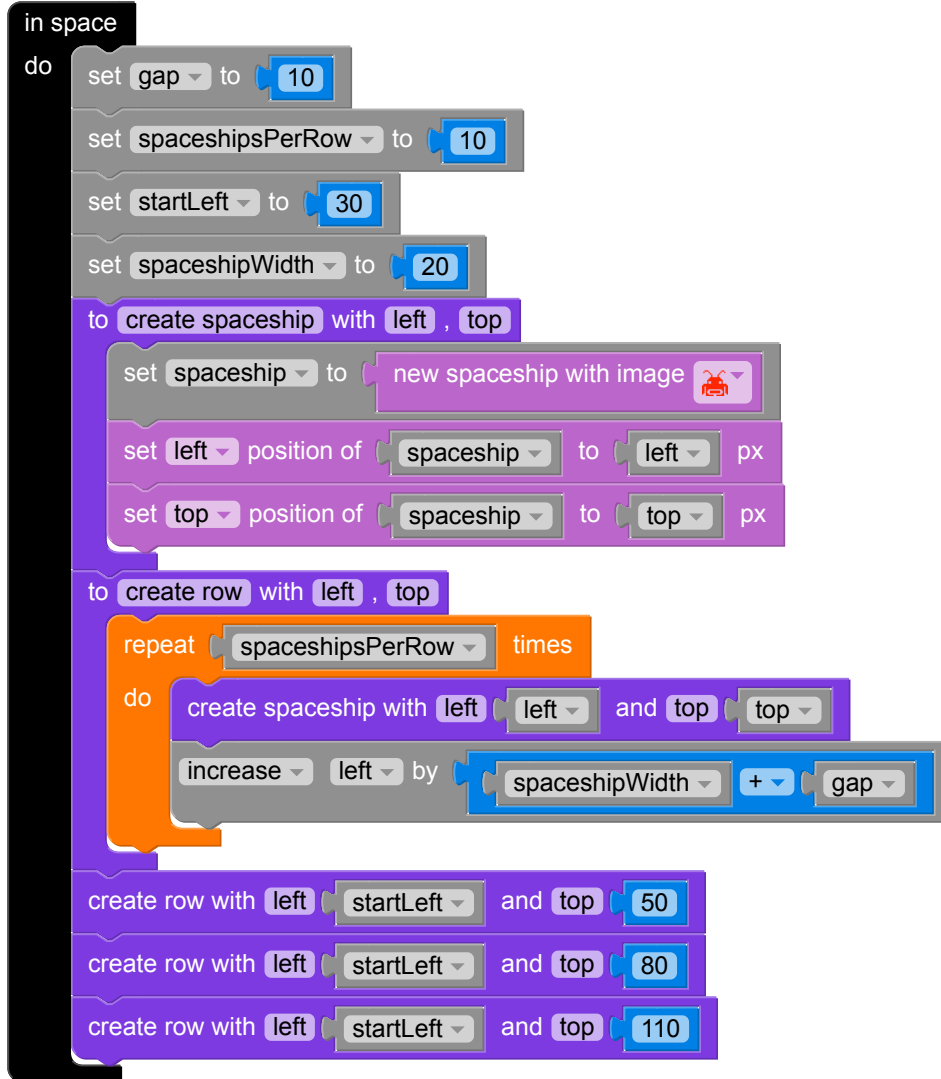
Build your program starting with the code already given.

The result should look something like this:
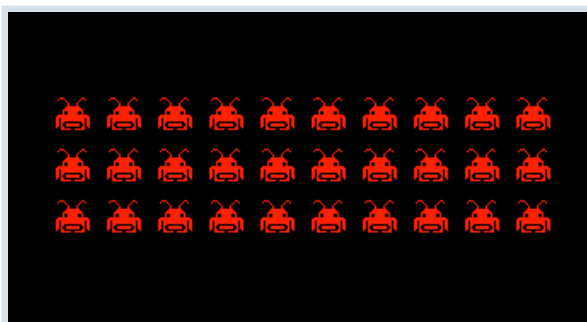
(https://groklearning-cdn.com/problems/kCWZLJGe9hNTbF5DAq3ZnJ/si-step4-3threerows.png)

Three rows of spaceships

You'll need

program.blockly

```
in space
do
    set gap to 10
    set spaceshipsPerRow to 10
    set startLeft to 30
    set spaceshipWidth to 20
    to create spaceship with left , top
        set spaceship to new spaceship with image 👾
        set left position of spaceship to left px
        set top position of spaceship to top px
    to create row with left , top
        create row with left startLeft and top 50

        repeat spaceshipsPerRow times
        do
            create spaceship with left left and top top
            increase left by spaceshipWidth + gap
```

Testing

☐ Testing the position of three rows of spaceships. Good Job! Your code has passed our test.

2.1.11. The moon of Xobron IV

Captain, we have arrived at the moon of Xobron IV, which is the halfway station on our journey to Earth.

Here, we will mine for precious resources for our 3D spaceship printer to make more spaceships.

0:00 / 0:09
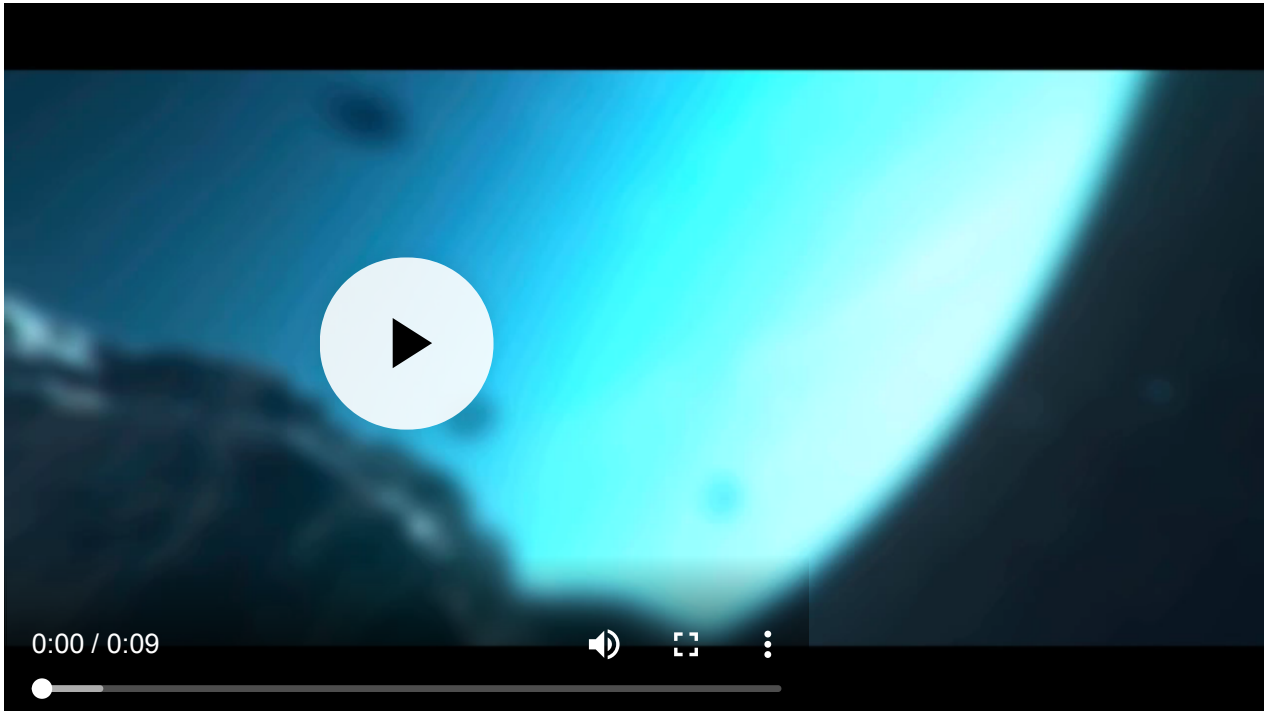
The Moon of Xobron IV (https://groklearning-cdn.com/modules/hSf9s4kEwcCGmAGcAWdiW3/XobronIV480p.mov).

2.1.12. Problem: Seven rows of spaceships

Captain, now that we have mined the resources, we can make even more spaceships.

You already know how to make three rows of spaceships. Now you can make **seven rows**

- The first row starts at `50` pixels from the top
- All rows start at `30` pixels from the left
- There is a gap of `10` pixels between each spaceship above, below and to the side

Build your program starting with the code already given.

The result should look something like this:
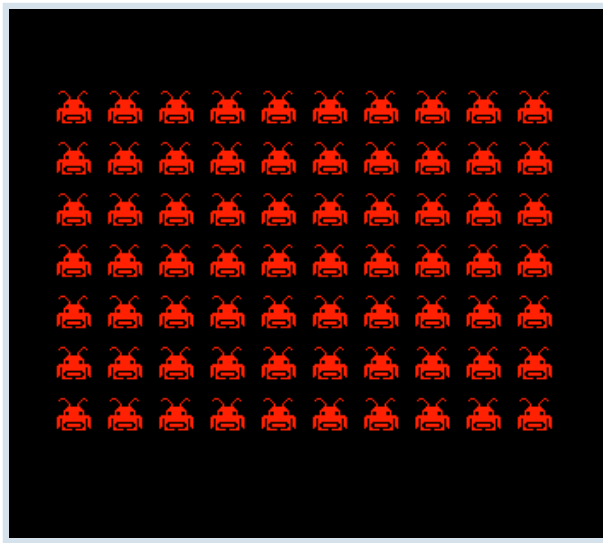
(https://groklearning-cdn.com/problems/ExhjWwLKGfaYk8pWqFN9yj/si-step5.png)

A seven-row grid of spaceships

You'll need

📄 program.blockly

```
in space
do
    set gap to 10
    set spaceshipsPerRow to 10
    set startLeft to 30
    set spaceshipWidth to 20
    to create spaceship with left , top
        set spaceship to new spaceship with image 👾
        set left position of spaceship to left px
        set top position of spaceship to top px

    to create row with left , top
        repeat spaceshipsPerRow times
        do  create spaceship with left left and top top
            increase left by spaceshipWidth + gap

    create row with left startLeft and top 50
    create row with left startLeft and top 80
    create row with left startLeft and top 110
```

Testing

☐ Well done. Test successful. Push ahead, captain.

2.1.13. Problem: Grid of spaceships ⌨

THE DOCTOR has suggested to place the code that makes the `create row with` function call inside another loop and place this into a new function `to create grid`

💡 Hints

We make three more variables. The number rows we want to build, the distance from the top to the first row of spaceships and the height of a spaceship.

```
set rowsOfSpaceships ▼ to    7
set startTop ▼ to    50
set spaceshipHeight ▼ to    20
```
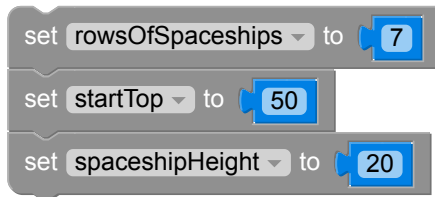
The vertical distance between the rows can be calculated as `spaceshipHeight+gap`. Make a loop that iterates through the number of rows and place it inside `to create grid`.

- Make `7` rows of spaceships
- The first row starts `50` pixels from the top
- All rows start at `30` pixels from the left
- There is a gap of `10` pixels between each spaceship above, below and to the side

The result should look something like this:

(https://groklearning-cdn.com/problems/ExhjWwLKGfaYk8pWqFN9yj/si-step5.png)

A seven-row grid of spaceships

💡 Hint - Click here to see the complete code

You'll need

program.blockly

```
in space
do
    set gap to 10
    set spaceshipsPerRow to 10
    set startLeft to 30
    set spaceshipWidth to 20
    set spaceshipHeight to 20
    to create spaceship with left , top
        set spaceship to new spaceship with image 👾
        set left position of spaceship to left px
        set top position of spaceship to top px

    to create row with left , top
        repeat spaceshipsPerRow times
        do  create spaceship with left left and top top
            increase left by spaceshipWidth + gap

    to create grid

    create grid
```
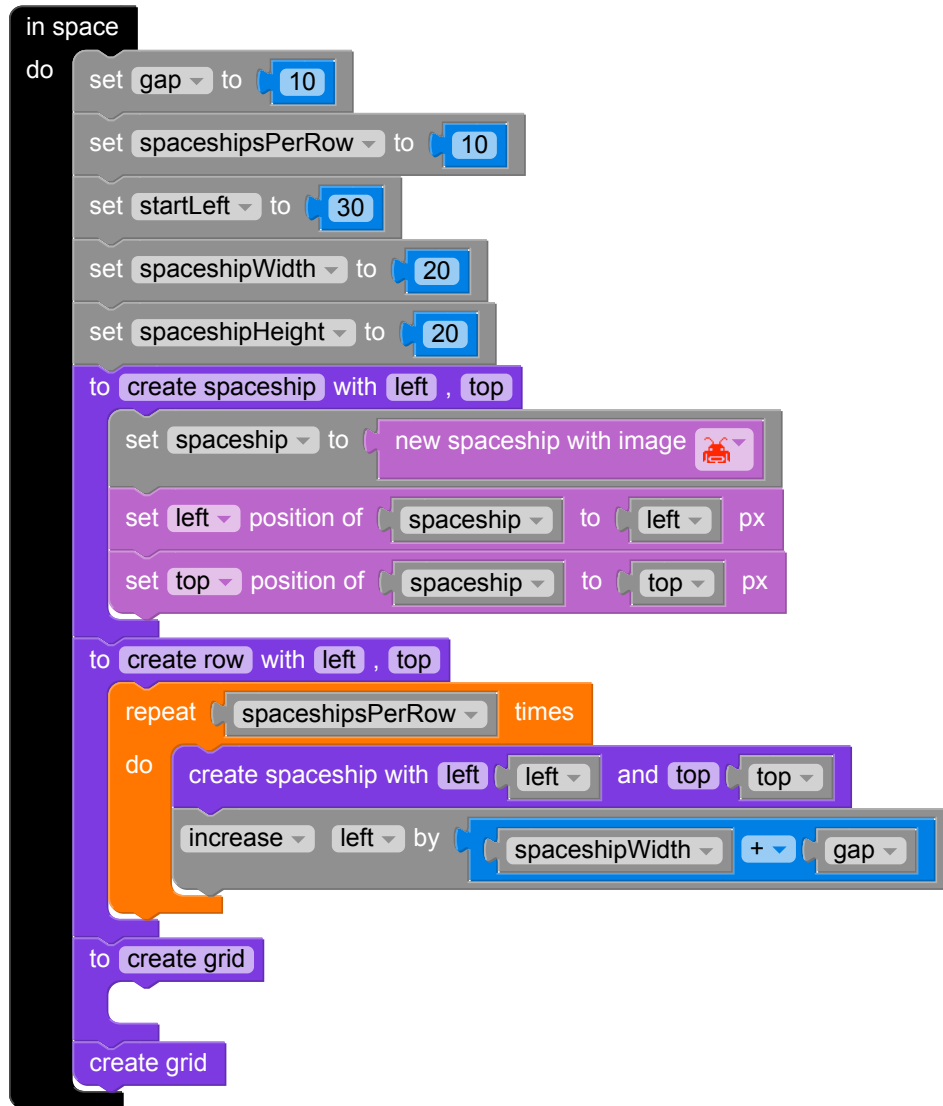
Testing

☐ Congratulations, Captain. Your code is perfect.

2.1.14. Problem: Grid of spaceships quiz ⌨

In which order do these functions need to be called to make a grid of spaceships ?

To solve this question, take a look at the generated JavaScript code from the previous mision. You find it on the left hand side of the Preview tab.

○ 1. createGrid()
2. createRow(left, top)
3. createSpaceship(left, top)

○ 1. createRow(left, top)
2. createGrid()
3. createSpaceship(left, top)

○ 1. createSpaceship(left, top)
2. createRow(left, top)
3. createGrid()

○ 1. createRow(left, top)
2. createGrid()
3. createSpaceship(left, top)

Testing

☐ That's right! We first call createGrid, which calls createRow, which in turn calls createSpaceship

2.1.15. Commodore

Captain, congratulations on the successful completion of the fleet building program. The Zargan High Command is very proud of your achievements.

The High Command hereby promotes you to the rank of:

Commodore

The people of planet Zarg congratulate you. A street in the capital city Zargantia has been named in your honour.

Long live the Zargan Empire.
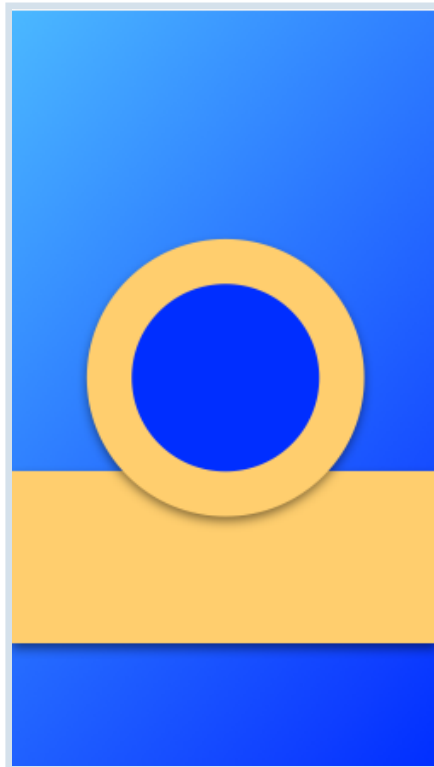
Commodore's Badge

3

STARDRIVE OPERATION

3.1. Stardrive operation

3.1.1. Mission briefing

Commodore, you have successfully created the Zargan space fleet.

Before we can engage Earth's Global Defence System, we have to make the fleet move in space.

Each spaceship is fitted with a stardrive that can propel the ship right, left, or towards the planet.

Your mission is to write a guidance system that controls each spaceship's stardrive.

The result will look something like this:

0:00 / 0:20
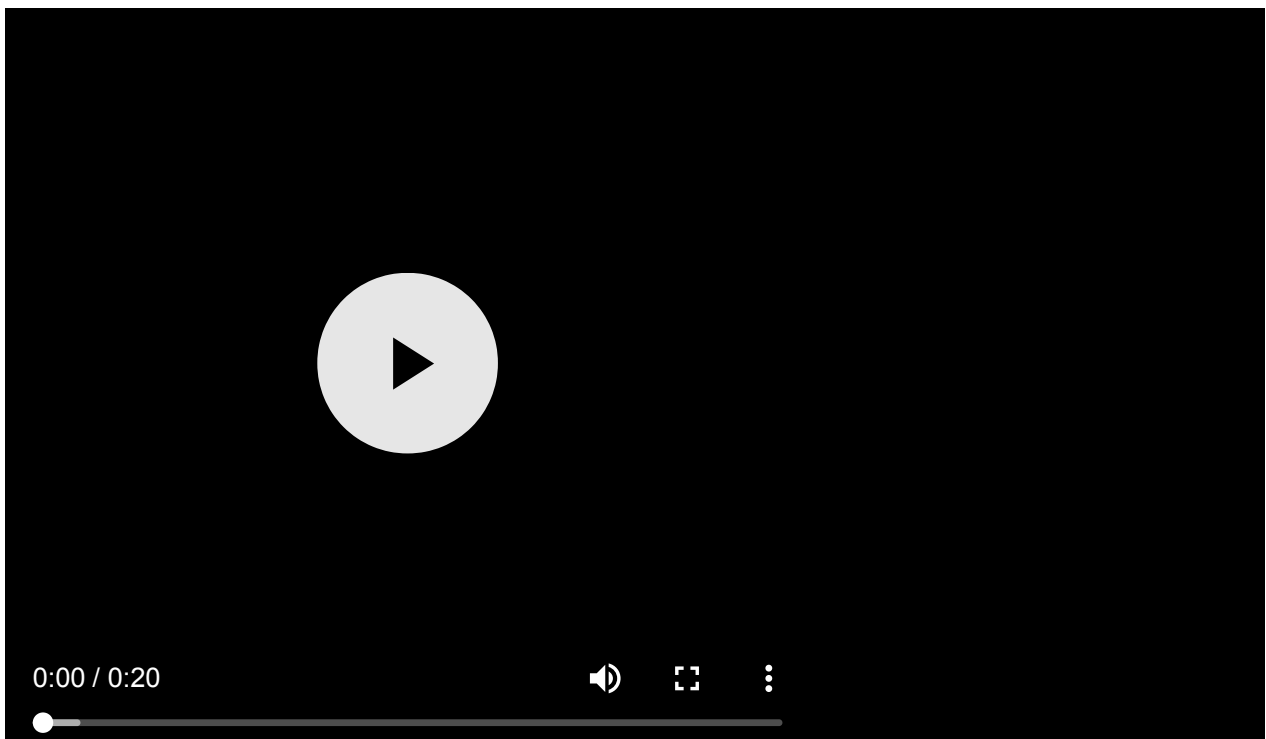
The Zargan Fleet: Attack Formation (https://groklearning-cdn.com/modules/icgk9aSDc69dzr4wzBGLgS/missionbriefing3sound.mp4).

3.1.2. Zeti Beta II

Commodore, we conduct our final preparations in the shelter of Zeti Beta II.

The planet's strong magnetic fields will shield us from Earth's sensors.

Two years ago, the High Command sent a spy satellite to check whether the earthlings know about this planet.

Luckily for us, they don't.

Helm reports that we are about to enter into orbit.

0:00 / 0:09
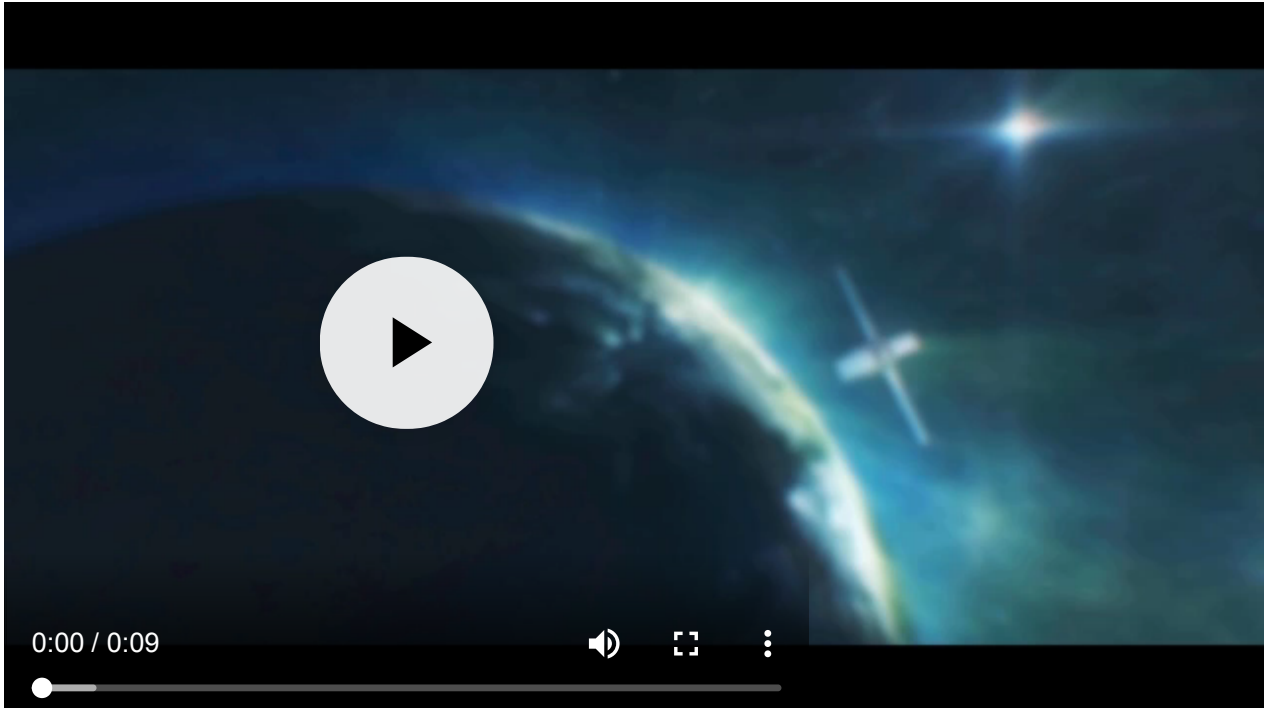
Zeti Beta II (https://groklearning-cdn.com/modules/5BgdhJM5CsQZhFRhr2tzi3/zeti-beta_ii-480p.mov).

3.1.3. Problem: Moving spaceships right ⌨

Commodore, we will first try to move spaceships to the right. In this mission, you need to update the spaceships' distance from the left.

- Make `5` rows of `10` spaceships.
- The first row starts `50` pixels from the top
- All rows start at `30` pixels from the left
- There is a gap of `10` pixels between each spaceship above, below and to the side

Step 1: Make a new function moveSpaceships

Step 2: Loop across all spaceships

Step 3: Change a spaceship's left position

Step 4: Animate

Step 5: Set the fleet in motion

The result should look something like this:

Spaceships moving to the right (https://groklearning-cdn.com/problems/jaUoPAEnmzv7ciD6cBABEk/spaceships-moving-right.mov).

Don't worry that the spaceships move off the screen. We will fix this in the next mission.

💡 **Hint - Click here to see the complete code**

You'll need

program.blockly

```
in space
do
    set gap to 10
    set startLeft to 30
    set startTop to 50
    set spaceshipWidth to 20
    set spaceshipHeight to 20
    set rowsOfSpaceships to 5
    set spaceshipsPerRow to 10

    to create spaceship with left , top
        set spaceship to new spaceship with image 👾
        set left position of spaceship to left px
        set top position of spaceship to top px

    to create row with left , top
        repeat spaceshipsPerRow times
        do  create spaceship with left left and top top
            increase left by spaceshipWidth + gap

    to create grid
        repeat rowsOfSpaceships times
        do  create row with left startLeft and top startTop
            increase startTop by spaceshipHeight + gap
```

Testing

☐ The fleet moves successfully to the right. Good Job! Your code has passed our test.

3.1.4. Problem: Moving spaceships right-left-.... ⌨

Commodore, we need to stop the fleet from moving off the screen on the right. To do this, we let the fleet travel to the right as before. But once it has travelled 100 pixels, we turn it around.

These are the steps:

Step 1: Make a new variable

Step 2: Use the new variable

Step 3: Make another variable

Step 4: Update Animate

The result should look something like this:
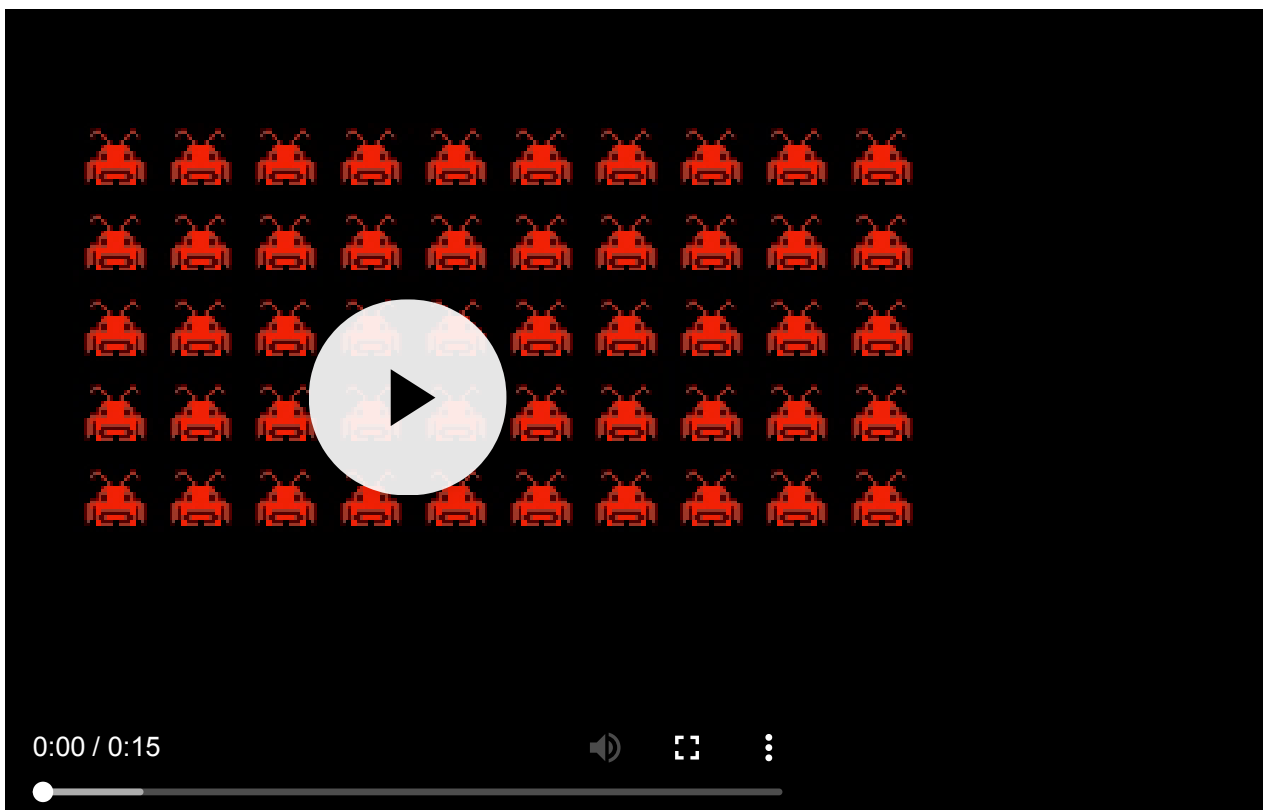
[Spaceships moving right-left-what ???? (https://groklearning-cdn.com/modules/9dskDRixVKnKEu5E8ZXkx3/spaceships-moving-right-left.mov)](https://groklearning-cdn.com/modules/9dskDRixVKnKEu5E8ZXkx3/spaceships-moving-right-left.mov)

It is unfortunate that our fleet now disappears on the left.

Move to the next mission, Commodore, to fix this.

💡 **Hint - Click here to see the complete code**

You'll need

[◇] program.blockly

```
in space
do
  set rowsOfSpaceships ▾ to  5
  set startTop ▾ to  50
  set gap ▾ to  10
  set spaceshipsPerRow ▾ to  10
  set startLeft ▾ to  30
  set spaceshipWidth ▾ to  20
  set spaceshipHeight ▾ to  20

  to create spaceship with left , top
    set ship ▾ to  new spaceship with image 👾▾
    set left ▾ position of  ship ▾  to  left ▾  px
    set top ▾ position of  ship ▾  to  top ▾  px

  to create row with left , top
    repeat  spaceshipsPerRow ▾  times
    do
      create spaceship with left  left ▾  and top  top ▾
      increase ▾ left ▾ by  spaceshipWidth ▾  + ▾  gap ▾

  to create grid
    repeat  rowsOfSpaceships ▾  times
    do
      create row with left  startLeft ▾  and top  startTop ▾
      increase ▾ startTop ▾ by  spaceshipHeight ▾  + ▾  gap ▾

  to move spaceships
    for each item ship ▾ in  all spaceships
    do
      change left ▾ position of  ship ▾  by  1  px

  to animate
    move spaceships
    then animate again

  create grid
  start animating
```

Testing

☐ Testing the animation of the fleet. Good Job! Your code has passed our test.

3.1.5. Problem: Moving spaceships right-left-right .... ⌨

Commodore, we want to stop the fleet from moving off the screen on the left. To do this, we let the fleet travel to the left as before. But once it has travelled 100 pixels, we turn it around (again).

There is really only one thing to do:

Step 1: Add a second decision-block

The result should look something like this:

0:00 / 0:06

Spaceships moving right-left-right. Perfect ! (https://groklearning-cdn.com/problems/3mfubX87hcZMqnirDLvWfZ/spaceships-oscillating.mov).

This cool algorithm sets xSpeed to 1 when the fleet is left and sets xSpeed to -1 when the fleet is right. Our fleet will keep moving right-left-right forever. xDistance grows when xSpeed is 1 and shrinks when xSpeed is -1. This way, xDistance stays in the range of 0 to 100.

We'll figure out how to move the fleet down in the next mission.

Keep pushing ahead Commodore, we are almost there.

💡 **Hint - Click here to see the complete code**

You'll need

<> **program.blockly**

```
in space
do
    set xDistance ▾ to [ 1 ]
    set xSpeed ▾ to [ 1 ]
    set rowsOfSpaceships ▾ to [ 5 ]
    set startTop ▾ to [ 50 ]
    set gap ▾ to [ 10 ]
    set spaceshipsPerRow ▾ to [ 10 ]
    set startLeft ▾ to [ 30 ]
    set spaceshipWidth ▾ to [ 20 ]
    set spaceshipHeight ▾ to [ 20 ]

    to create spaceship with [ left ] , [ top ]
        set ship ▾ to [ new spaceship with image 👾 ▾ ]
        set left ▾ position of [ ship ▾ ] to [ left ▾ ] px
        set top ▾ position of [ ship ▾ ] to [ top ▾ ] px

    to create row with [ left ] , [ top ]
        repeat [ spaceshipsPerRow ▾ ] times
        do  create spaceship with [ left [ left ▾ ] and top [ top ▾ ]
            increase ▾ [ left ▾ ] by [ spaceshipWidth ▾ [ + ▾ ] gap ▾ ]

    to create grid
        repeat [ rowsOfSpaceships ▾ ] times
        do  create row with [ left [ startLeft ▾ ] and top [ startTop ▾ ]
            increase ▾ [ startTop ▾ ] by [ spaceshipHeight ▾ [ + ▾ ] gap ▾ ]

    to move spaceships
        for each item [ ship ▾ ] in [ all spaceships ]
        do  change [ left ▾ ] position of [ ship ▾ ] by [ xSpeed ▾ ] px

    to animate
        if [ xDistance ▾ [ = ▾ ] [ 100 ] ]
        do  set xSpeed ▾ to [ -1 ]

        increase ▾ [ xDistance ▾ ] by [ xSpeed ▾ ]
        move spaceships
```
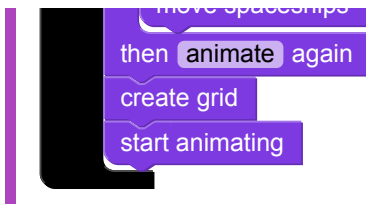
Testing

☐ Fantastic, the fleet is moving right-left-right as expected

3.1.6. Problem: Moving spaceships down ⌨

Commodore, as a final step we now move the fleet down, towards Earth. Each time the fleet has travelled 100 pixels, it moves down by 10 pixels.

These are the steps:

Step 1: Make a new variable

Step 2: Add a change to the top position

Step 3: Extend the decision-blocks

Step 4: Don't let the fleet crash !

That's it. Each time the fleet has travelled left or right, the variable ySpeed is set to 10 and `moveSpaceships` will, just then, move the spaceships in x-direction and also in y-direction.
The result should look like this:

0:00 / 0:20
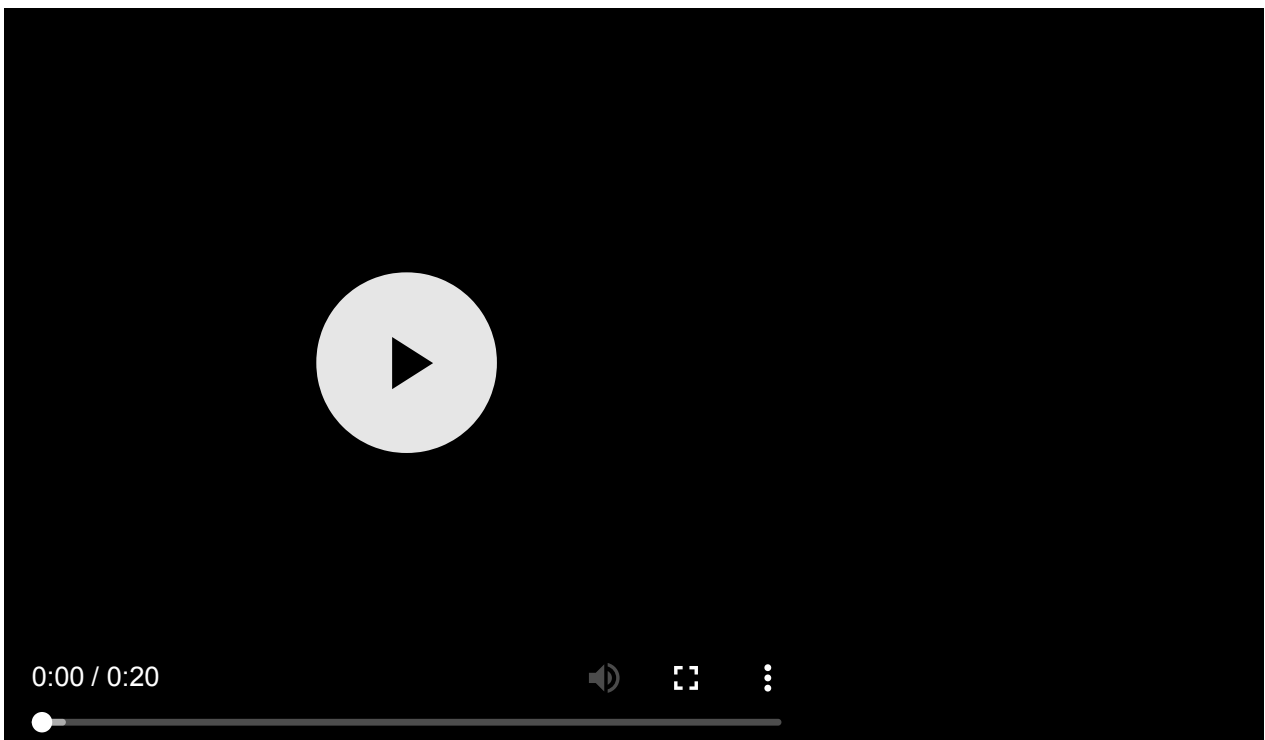
The Zargan Fleet: Attack Formation (https://groklearning-cdn.com/modules/VVVKujr4gqwwRYnrSMxZ5n/MissionBriefing3.mov).

Congratulations, Commodore. You have successfully programmed the stardrive operating system of our spaceships. We are now able to maneuver them in space

💡 **Hint - Click here to see the complete code**

You'll need

<> program.blockly

<> program.blockly

```
in space
do
    set xDistance to 1
    set xSpeed to 1
    set ySpeed to 0
    set rowsOfSpaceships to 5
    set startTop to 50
    set gap to 10
    set spaceshipsPerRow to 10
    set startLeft to 30
    set spaceshipWidth to 20
    set spaceshipHeight to 20

    to create spaceship with left , top
        set ship to new spaceship with image 👾
        set left position of ship to left px
        set top position of ship to top px

    to create row with left , top
        repeat spaceshipsPerRow times
        do create spaceship with left left and top top
           increase left by spaceshipWidth + gap

    to create grid
        repeat rowsOfSpaceships times
        do create row with left startLeft and top startTop
           increase startTop by spaceshipHeight + gap

    to move spaceships
        for each item ship in all spaceships
        do change left position of ship by xSpeed px
           change top position of ship by ySpeed px

    to animate
        if xDistance = 100
        do set xSpeed to 1
```

Testing

☐ Testing the animation of the fleet. Good Job! Your code has passed our test.

3.1.7. Problem: Moving Spaceships quiz

What will this code do?

```
to animate
  for each item ship ▼ in ( all spaceships
  do  change left ▼ position of ( ship ▼ ) by ( 1 ) px
      change top ▼ position of ( ship ▼ ) by ( 1 ) px
  then animate again
```

○ Move spaceships in a diagonal line from top-left to bottom-right.

○ Make spaceships jump around randomly

○ Move spaceships in a horizontal line from right to left

○ Nothing. Spaceships stay where they are.

Testing

☐ That's right!

3.1.8. The cover is blown

Commodore, we have been spotted by Earth's early warning station deep inside the asteroid belt between Mars and Jupiter.
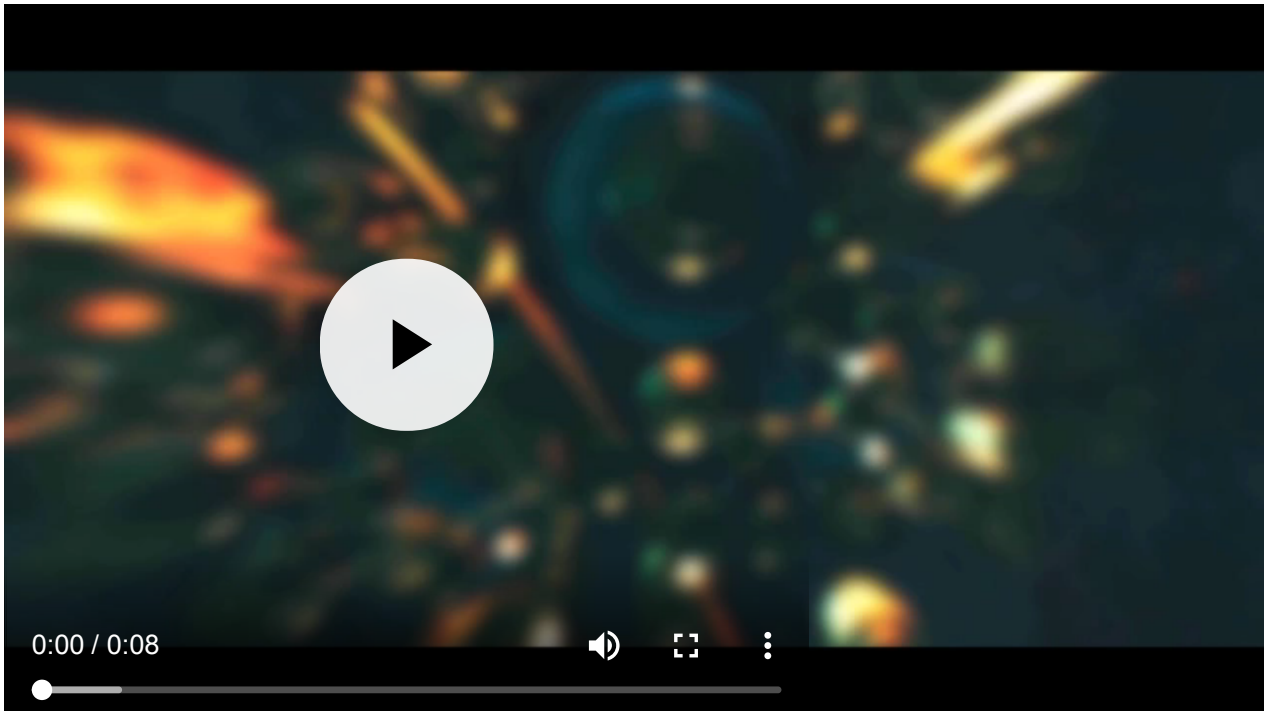
Asteroid Base (https://groklearning-cdn.com/modules/jzzWerNY6FHE4MXatyobH3/asteroidbase.mov).

Earth knows we're coming. We have to move fast. You are authorised to use the experimental wormhole drive that will catapult us straight to Earth. Beware, the drive hasn't been tested yet. But we have no other choice.
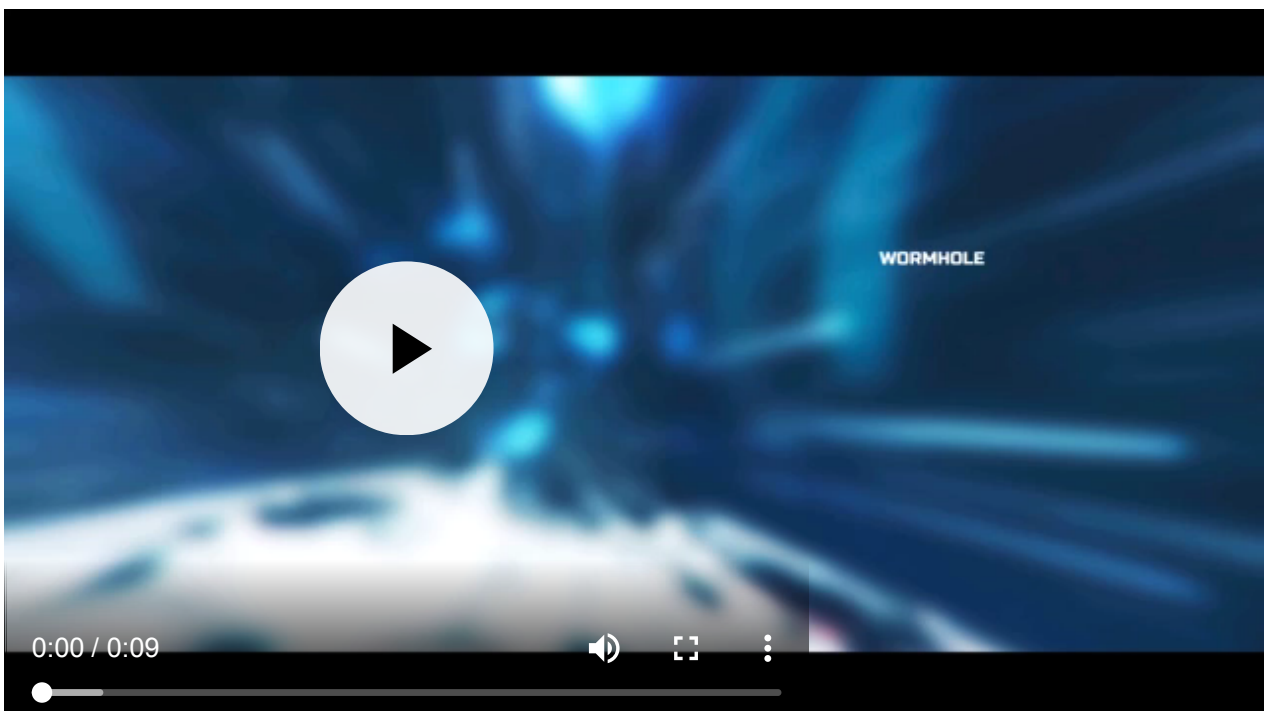
Wormhole Drive (https://groklearning-cdn.com/modules/z9MmVGrz32vqyqcX3koQLa/wormhole.mov).

3.1.9. Problem: Solar system quiz

What is the correct order of planets in the solar system as we travel towards the sun?

○

○

○

○
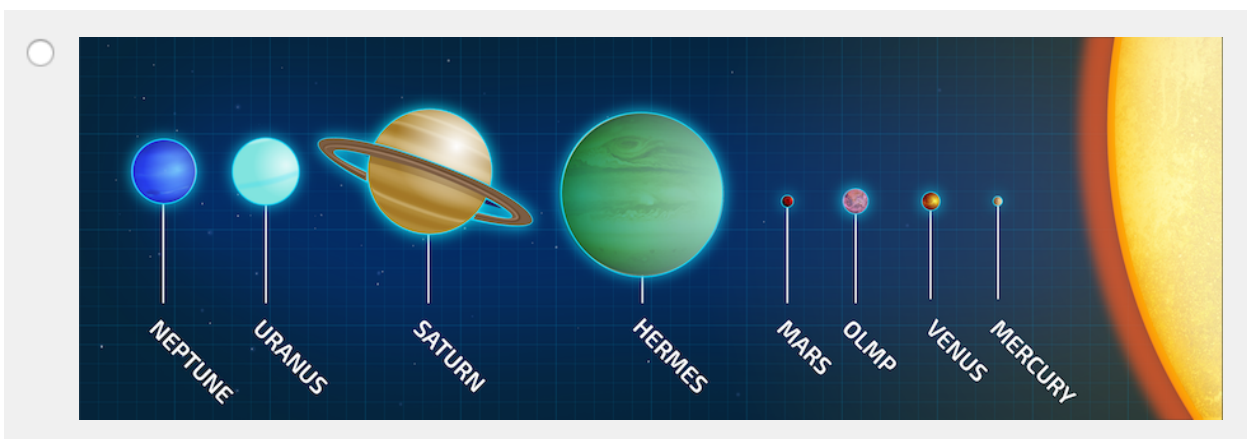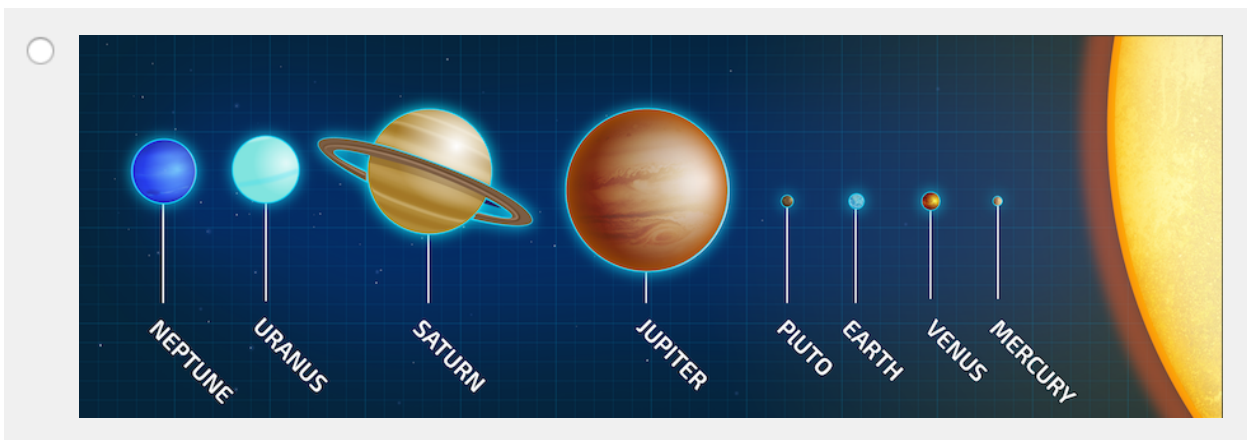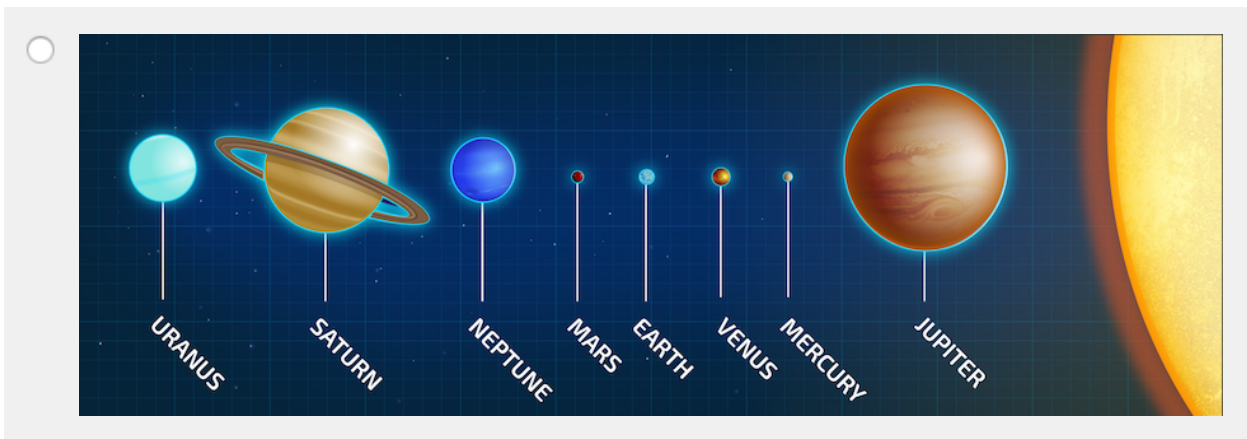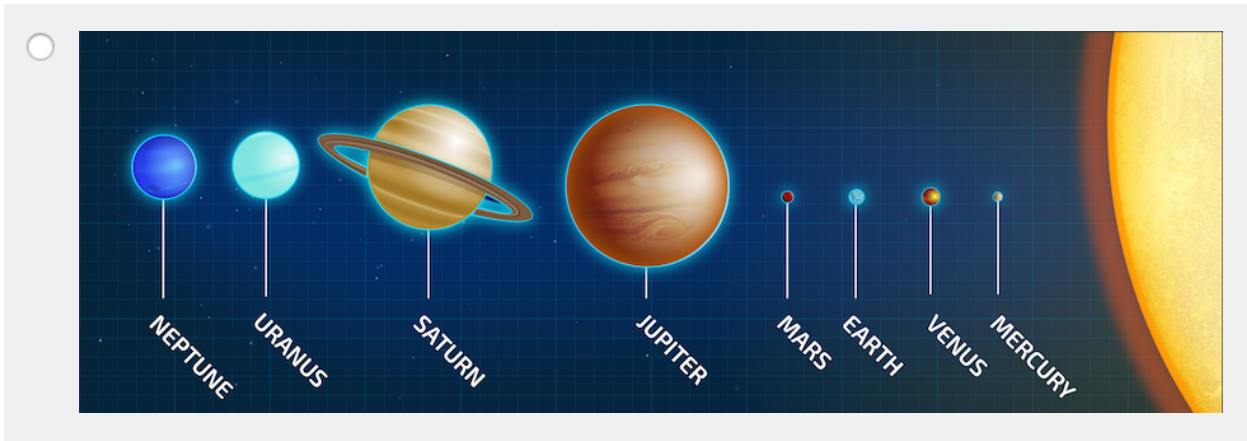
Testing

☐ Yes, this is the right order of planets.

3.1.10. Earth

Commodore, we have finally reached the blue planet, Earth.

Look at all that water.

We are presently in orbit above Mexico. Ahead of us is the Gulf of California. We continue west to our landing coordinates in the Pacific Ocean, at the east coast of the continent of Australia.

0:00 / 0:24
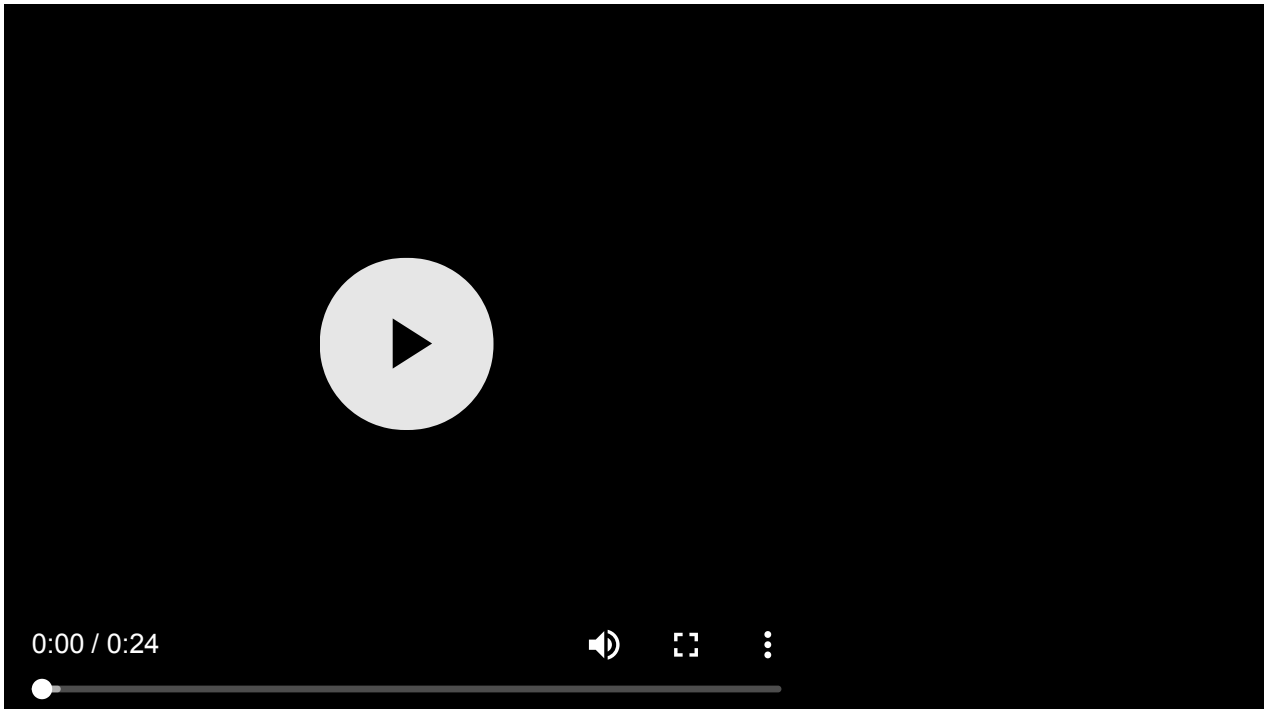
Earth (https://groklearning-cdn.com/modules/SnkK98tT7QPLgef4gNAxQU/earth480p.mov).

3.1.11. Rear Admiral

Commodore, congratulations on the successful completion of the fleet motion program and well done for reaching Earth. The Zargan High Command is very proud of your achievements.

The High Command hereby promotes you to the rank of:

Rear Admiral

The government of planet Zarg congratulates.

A city in the province of Zargon has been named in your honour.

Long live the Zargan Empire.

Rear Admiral's Badge

4

THE BATTLE FOR EARTH

4.1. The battle for Earth

4.1.1. Mission briefing

Rear Admiral, the fleet reports ready for the invasion of Earth.

Your mission is to successfully land as many spaceships as possible on Earth's Pacific Ocean off the coast of the continent of Australia.

Once landed, the spaceships will deploy a teleportation device with which we can beam Earth's water into the mothership waiting in Orbit.

Be warned, the earthlings expect us and have activated the Global Defence System (GDS).

0:00 / 0:12

Red Alert ... Global Defence System Activated (https://groklearning-cdn.com/modules/Aw8MiiXbZAkDbJPZyr26F5/gdsactivated480p.mov).

4.1.2. Problem: Battle for Earth ⌨

Rear Admiral, you are about to engage Earth's global defence system.

Two counters at the top of the page will inform you of your remaining fleet size and distance to the landing spot. You have to land at least one spaceship to win.

Our spaceships are invisible to Earth's radar. All Earth can do is to shoot rockets blindly into space. Beware though - they are lethal.

You can experiment with all sorts of configurations to your fleet, such as the number of spaceships, gaps between them, etc. To get this challenged marked, though, you are not allowed more than 50 spaceships, a ySpeed of no greater than 10, and xDistanceMin has to be at least 300. All these values are set by default in the sample code provided.

The result should look something like this:
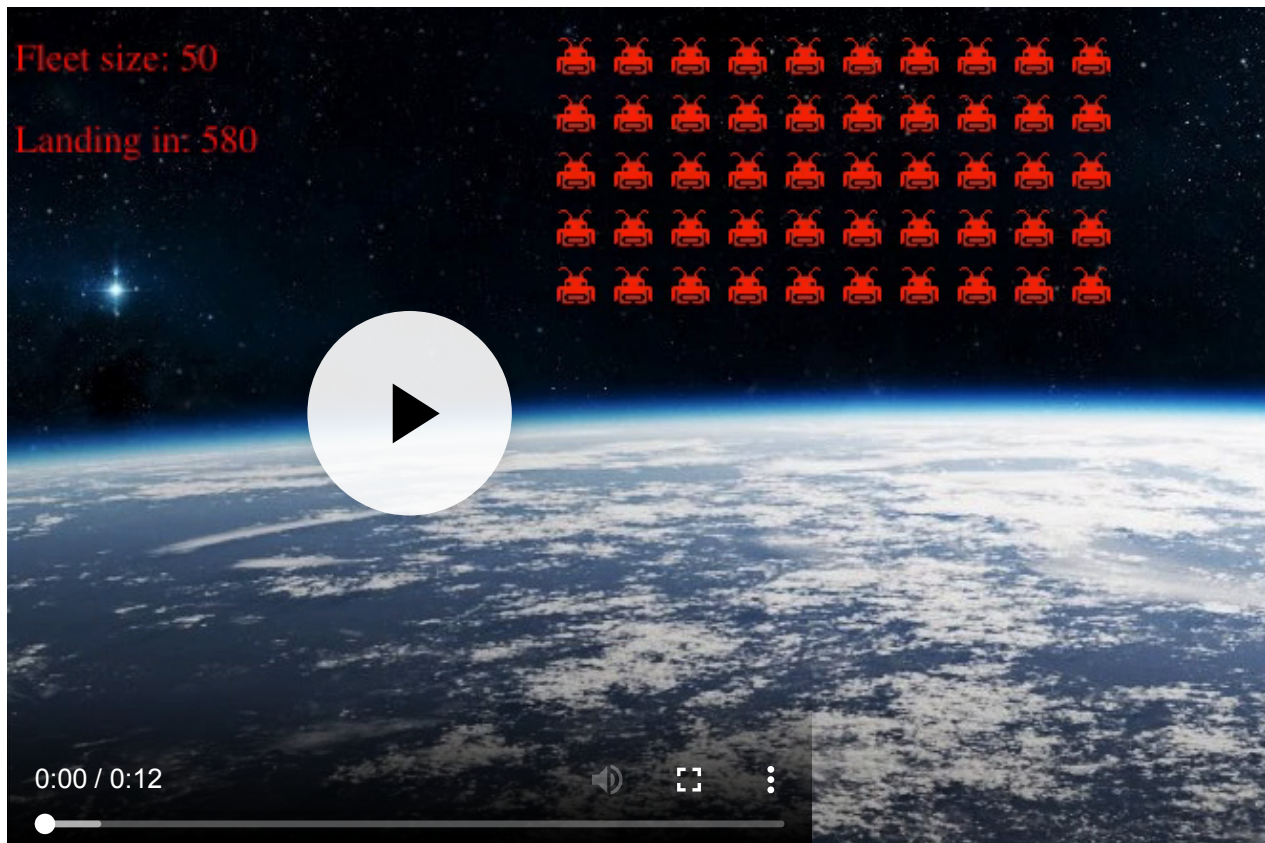
Battle for Earth (https://groklearning-cdn.com/modules/7788e3Ah3YVzstVhrfPrEL/battleForEarth.mov).

You'll need

program.blockly

```
in the battle for earth
do
    set ySpeedMax to [ 10 ]
    set xDistanceMin to [ 300 ]
    set xDistance to [ 1 ]
    set xSpeed to [ 1 ]
    set ySpeed to [ 0 ]
    set rowsOfSpaceships to [ 5 ]
    set spaceshipsPerRow to [ 10 ]
    set startTop to [ 50 ]
    set startLeft to [ 150 ]
    set gap to [ 10 ]
    set spaceshipWidth to [ 20 ]
    set spaceshipHeight to [ 20 ]

    to create spaceship with left , top
        set ship to ( new spaceship with image 👾 )
        set left position of ( ship ) to ( left ) px
        set top position of ( ship ) to ( top ) px

    to create row with left , top
        repeat ( spaceshipsPerRow ) times
        do  create spaceship with left ( left ) and top ( top )
            increase left by ( spaceshipWidth + gap )

    to create grid
        repeat ( rowsOfSpaceships ) times
        do  create row with left ( startLeft ) and top ( startTop )
            increase startTop by ( spaceshipHeight + gap )

    to move spaceships
        for each item ship in ( all spaceships )
        do  change left position of ( ship ) by ( xSpeed ) px
            change top position of ( ship ) by ( ySpeed ) px

    to animate
```

Testing

☐ Testing the animation of the fleet.

4.1.3. Admiral of the Fleet

Congratulations on the successful invasion of Earth. Thanks to your efforts and valour planet Zarg has been saved. The water that you have secured from Earth has been used to restore the Zargan climate.

Luckily, Planet Zarg is 20 times smaller than Earth, so we didn't have to take all that much water from Earth. As a result, Earth's precious ecosystem is not very much affected.

The Zargan High Command is very proud of your achievement and hereby promotes you to the highest rank in the Zargan Spacefleet:

Admiral of the Fleet

The government and all people of planet Zarg congratulate.

You have been awarded a brand new command to zarg-form Earth's neighbour, Mars, to make it hospitable for future generations of Zargans .

Long live the Zargan Empire.
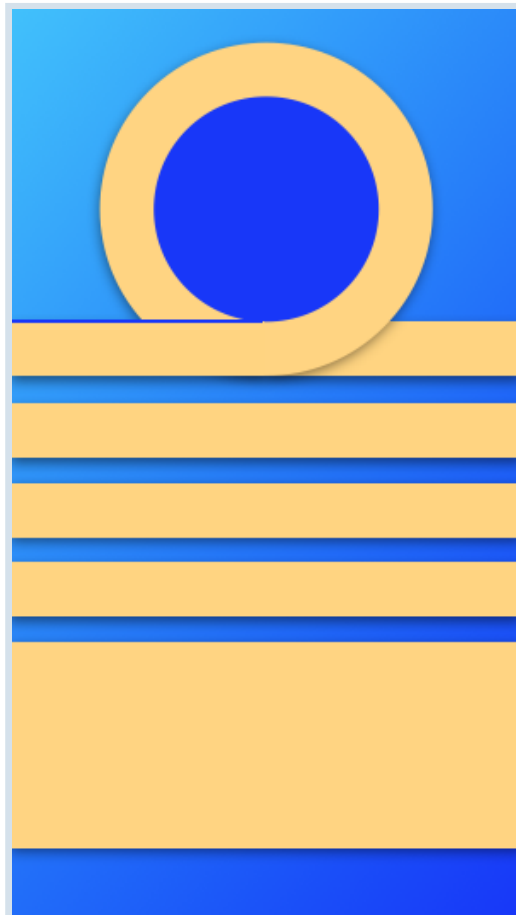
Admiral's Badge

4.1.4. Let's Celebrate

Planet Zarg is celebrating your victory with huge fireworks. Enjoy!

Live broadcast from planet Zarg (https://groklearning-cdn.com/modules/9NZ5v38XB8NCaLKGtnTCbN/Fireworks480p.mov).

4.1.5. Problem: The curtain rises ⌨

Now that you have completed this course we wanted to show you some of the stuff that went on behind the scenes.

We have now made visible the files that supported your program code. Here is the list of the files and a short description of what they do.

- **index.html** is the starting point. It is loaded by the browser first. It contains the JavaScript code that is generated from your blocks. It also makes reference to the files below:
- **stylesheet.css** is a cascading stylesheet document that makes our output look nicer. For example, it sets the Earth background image and the colour and size of the fields.
- **gds.js** is the code that runs the Global Defence System.
- The files **spaceship1..8** contain the absolute links to the corresponding spaceship images. With these files, we can access our spaceships by just calling, e.g. spaceship2.png, rather than the full URL

We encourage you to check out the code, make changes as you see fit and observe what happens. You can't break anything permanently, so have a go and explore.

You'll need

program.blockly

```xml
<xml xmlns="http://www.w3.org/1999/xhtml">
  <variables>
    <variable type="" id="Py}XFC1]}(G(+7O{saE[">ySpeedMax</variable>
    <variable type="" id="O.~BE,r-}NwmO*Rkx$Cz">xDistanceMin</variable>
    <variable type="" id="$1Qtq3#|uZMhwK~Lj=I?">xDistance</variable>
    <variable type="" id="`zcLIO5e8k|Pmr4a:3eb">xSpeed</variable>
    <variable type="" id=",1v7(;{NZg+n{`}7GMa|">ySpeed</variable>
    <variable type="" id="1o.q)a3NE,ReBt/~;@b}">rowsOfSpaceships</variable>
    <variable type="" id="jPh?~/{+kch`qq$3`71]">spaceshipsPerRow</variable>
    <variable type="" id="Y0^:Lc2_e[D#it9}?QP+">startTop</variable>
    <variable type="" id="5tEM+ySFs27/DXl$ZI]I">startLeft</variable>
    <variable type="" id="U5rMHL@G%msUsL=tz0e(">gap</variable>
    <variable type="" id="[d%+WCw2L8%X8cO05yfT">spaceshipWidth</variable>
    <variable type="" id="CY;-+UP;Fk#/H$SPcIpk">spaceshipHeight</variable>
    <variable type="" id="n)7x*5TKOVPMcP)Y;w7O">ship</variable>
    <variable type="" id=".xuA8NU[Y9:xk3}F(|T)">left</variable>
    <variable type="" id="Dy|/t1Re,)OL2jd@;bA3">top</variable>
  </variables>
  <block type="space_invaders_in_space_gds_script_element" id="SEE%-Yk7YNw[lCUdn{b~">
    <statement name="CONTENT">
      <block type="variables_set" id="0`XP}$/,X]MR~Fu3,$CE">
        <field name="VAR" id="Py}XFC1]}(G(+7O{saE[" variabletype="">ySpeedMax</field>
        <value name="VALUE">
          <block type="math_number" id="xRE7-hgW$[r^x#RTU1pb">
            <field name="NUM">10</field>
          </block>
        </value>
        <next>
          <block type="variables_set" id="a!H*BJqaPgZRwVVw:{Qf">
            <field name="VAR" id="O.~BE,r-}NwmO*Rkx$Cz" variabletype="">xDistanceMin<
            <value name="VALUE">
              <block type="math_number" id="3.J$i1t,GVF.pAP{a.rQ">
                <field name="NUM">300</field>
              </block>
            </value>
            <next>
              <block type="variables_set" id="RWn`!5@Sn2Xe%]08RHMW">
                <field name="VAR" id="$1Qtq3#|uZMhwK~Lj=I?" variabletype="">xDistance
                <value name="VALUE">
                  <block type="math_number" id="Z3{L|N|%`q/YE%2.sAr$">
                    <field name="NUM">1</field>
                  </block>
                </value>
                <next>
                  <block type="variables_set" id="PF[uBZ4u:Fi,/%T(2SjX">
                    <field name="VAR" id="`zcLIO5e8k|Pmr4a:3eb" variabletype="">xSpee
                    <value name="VALUE">
                      <block type="math_number" id=")fIOuc`M?GL6c@-ZG.JP">
                        <field name="NUM">1</field>
                      </block>
                    </value>
                    <next>
                      <block type="variables_set" id=",!-%?PF~MFdsDG,!sFl*">
                        <field name="VAR" id=",1v7(;{NZg+n{`}7GMa|" variabletype="">y
                        <value name="VALUE">
                          <block type="math_number" id="mZ?O;a@RL!A_ElC6s!VM">
                            <field name="NUM">0</field>
                          </block>
                        </value>
```
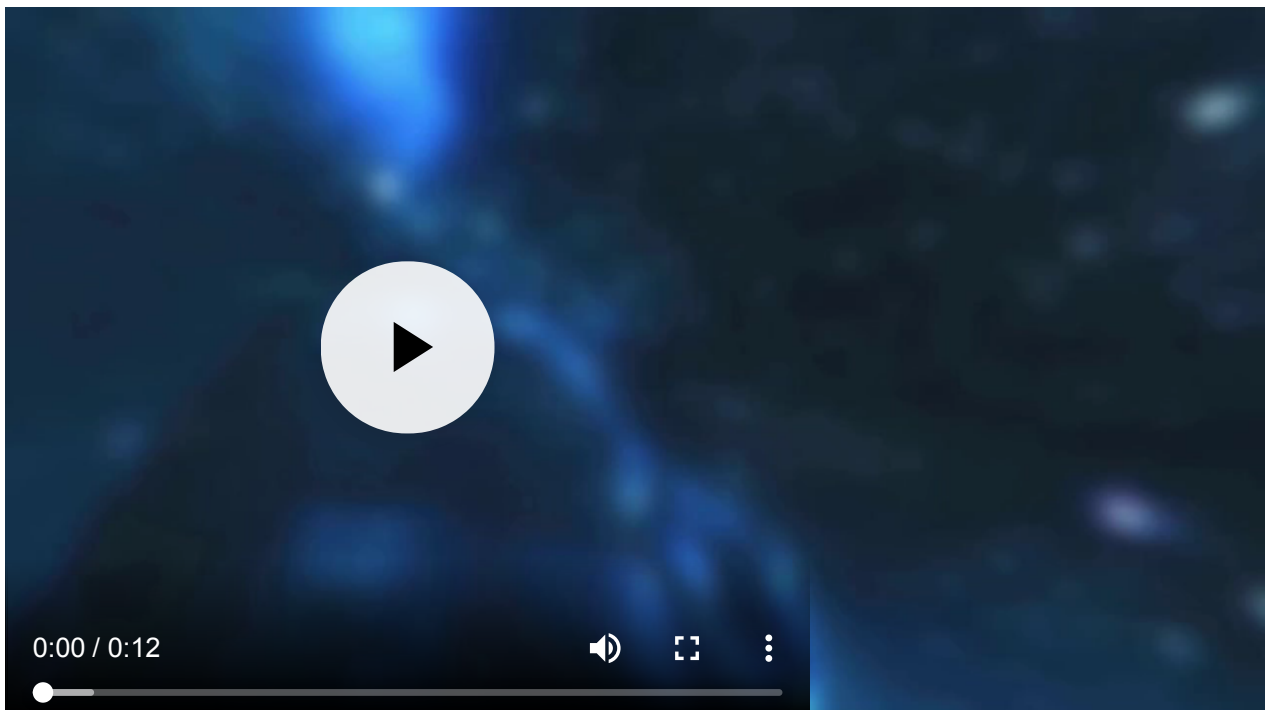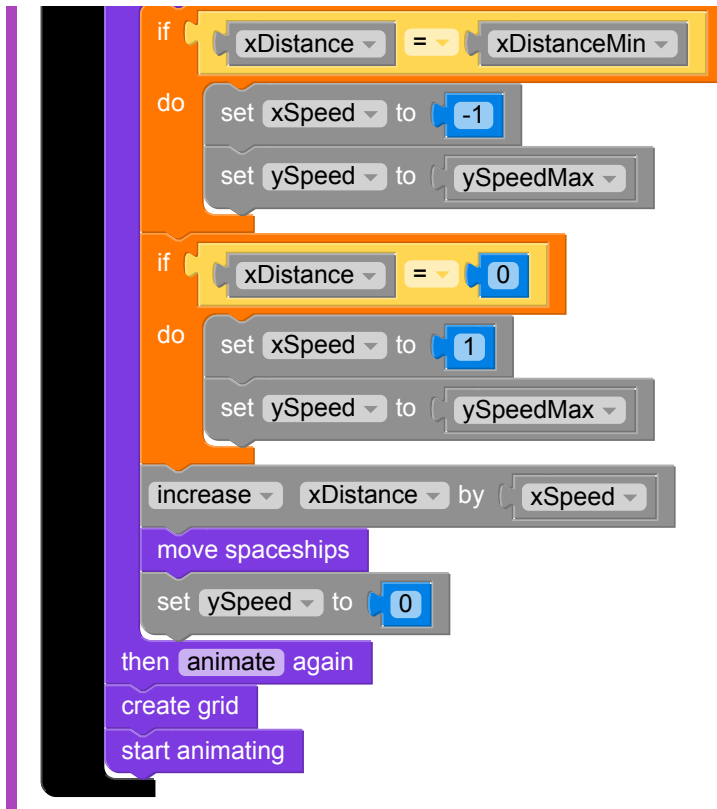
```xml
<next>
  <block type="variables_set" id="Q:q,yopnNTCD#d_T9{2{">
    <field name="VAR" id="1o.q)a3NE,ReBt/~;@b}" variabletype=
    <value name="VALUE">
      <block type="math_number" id="mDJ9+i{Vj.|#1,EcY6OH">
        <field name="NUM">5</field>
      </block>
    </value>
    <next>
      <block type="variables_set" id="m#d-/=8V=yX_FA/qcYGD">
        <field name="VAR" id="jPh?~/{+kch`qq$3`71]" variablet
        <value name="VALUE">
          <block type="math_number" id="-LoYS9mkoixZvZBV];o3"
            <field name="NUM">10</field>
          </block>
        </value>
        <next>
          <block type="variables_set" id="HjX-JoN.jH6e]tq#znP
            <field name="VAR" id="Y0^:Lc2_e[D#it9}?QP+" varia
            <value name="VALUE">
              <block type="math_number" id="^MiFfy_9:^dGlt5r*
                <field name="NUM">50</field>
              </block>
            </value>
            <next>
              <block type="variables_set" id=";FYDbwJO+U??@PN
                <field name="VAR" id="5tEM+ySFs27/DXl$ZI]I" v
                <value name="VALUE">
                  <block type="math_number" id="r,z6~t:kpD(gM
                    <field name="NUM">150</field>
                  </block>
                </value>
                <next>
                  <block type="variables_set" id=",4X[3yR/D/L
                    <field name="VAR" id="U5rMHL@G%msUsL=tz0e
                    <value name="VALUE">
                      <block type="math_number" id="q$7xU(Vmz
                        <field name="NUM">10</field>
                      </block>
                    </value>
                    <next>
                      <block type="variables_set" id="dXWc;Gd
                        <field name="VAR" id="[d%+WCw2L8%X8cO
                        <value name="VALUE">
                          <block type="math_number" id="4F_GZ
                            <field name="NUM">20</field>
                          </block>
                        </value>
                        <next>
                          <block type="variables_set" id="Q6H
                            <field name="VAR" id="CY;-+UP;Fk#
                            <value name="VALUE">
                              <block type="math_number" id="h
                                <field name="NUM">20</field>
                              </block>
                            </value>
                            <next>
                              <block type="space_invaders_def
                                <field name="NAME">create spa
```

```xml
<field name="LEFT">left</fiel
<field name="TOP">top</field>
<statement name="STACK">
  <block type="variables_set"
    <field name="VAR" id="n)7
    <value name="VALUE">
      <block type="space_inva
        <field name="URL">spa
      </block>
    </value>
    <next>
      <block type="space_inva
        <field name="POSITION
        <value name="ELEMENT"
          <block type="variab
            <field name="VAR"
          </block>
        </value>
        <value name="PIXELS">
          <block type="variab
            <field name="VAR"
          </block>
        </value>
        <next>
          <block type="space_
            <field name="POSI
            <value name="ELEM
              <block type="va
                <field name="
              </block>
            </value>
            <value name="PIXE
              <block type="va
                <field name="
              </block>
            </value>
          </block>
        </next>
      </block>
    </next>
  </block>
</statement>
<next>
  <block type="space_invaders
    <field name="NAME">create
    <field name="LEFT">left</
    <field name="TOP">top</fi
    <statement name="STACK">
      <block type="loops_repe
        <value name="TIMES">
          <block type="variab
            <field name="VAR"
          </block>
        </value>
        <statement name="DO">
          <block type="space_
            <field name="LEFT
            <field name="TOP"
            <value name="LEFT
```

```xml
            <block type="va
               <field name="
            </block>
         </value>
         <value name="TOP"
            <block type="va
               <field name="
            </block>
         </value>
         <next>
            <block type="va
               <field name="
               <field name="
               <value name="
                  <block type
                     <field na
                     <value na
                        <block
                           <fiel
                        </block
                     </value>
                     <value na
                        <block
                           <fiel
                        </block
                     </value>
                  </block>
               </value>
            </block>
         </next>
      </block>
   </statement>
</block>
</statement>
<next>
   <block type="space_inva
      <field name="NAME">cr
      <statement name="STAC
         <block type="loops_
            <value name="TIME
               <block type="va
                  <field name="
               </block>
            </value>
            <statement name="
               <block type="sp
                  <field name="
                  <field name="
                  <value name="
                     <block type
                        <field na
                     </block>
                  </value>
                  <value name="
                     <block type
                        <field na
                     </block>
                  </value>
                  <next>
```

```
<block type
  <field na
  <field na
  <value na
    <block
      <fiel
      <valu
        <bl
          <
        </b
      </val
      <valu
        <bl
          <
        </b
      </val
    </block
  </value>
</block>
</next>
</block>
</statement>
</block>
</statement>
<next>
<block type="space_
  <field name="NAME
  <statement name="
    <block type="lo
      <field name="
      <value name="
        <block type
      </value>
      <statement na
        <block type
          <field na
          <value na
            <block
              <fiel
            </block
          </value>
          <value na
            <block
              <fiel
            </block
          </value>
          <next>
            <block
              <fiel
              <valu
                <bl
                  <
                </b
              </val
              <valu
                <bl
                  <
                </b
              </val
```

```
                                </block
                            </next>
                        </block>
                    </statement>
                </block>
            </statement>
            <next>
                <block type="sp
                    <field name="
                    <field name="
                    <statement na
                        <block type
                            <value na
                                <block
                                    <fiel
                                    <valu
                                        <bl
                                            <
                                        </b
                                    </val
                                    <valu
                                        <bl
                                            <
                                        </b
                                    </val
                                </block
                            </value>
                            <statemen
                                <block
                                    <fiel
                                    <valu
                                        <bl
                                            <
                                        </b
                                    </val
                                    <next
                                        <bl
                                            <
                                            <


                                            <
                                        </b
                                    </nex
                                </block
                            </stateme
                            <next>
                                <block
                                    <valu
                                        <bl
                                            <
                                            <



                                            <
                                            <
```

```
                    <
                  </b
                 </val
                 <stat
                  <bl
                   <
                   <



                   <
                   <








                    <
                  </b
                 </sta
                 <next
                  <bl
                   <
                   <
                   <




                   <
                   <














                    <
                  </b
                 </nex
                </block
               </next>
              </block>
            </statement>
            <next>
              <block type
                <next>
                  <block
                </next>
```

```
                                          </block>
                                        </next>
                                      </block>
                                    </next>
                                  </block>
                                </next>
                              </block>
                            </next>
                          </block>
                        </next>
                      </block>
                    </next>
                  </block>
                </next>
              </block>
            </next>
          </block>
        </next>
      </block>
    </next>
  </block>
            </next>
          </block>
        </next>
      </block>
    </next>
  </block>
            </next>
          </block>
        </next>
      </block>
    </next>
  </block>
            </next>
          </block>
        </next>
      </block>
    </next>
  </block>
        </statement>
      </block>
  </block>
</xml>
```

</> **gds.js**

```
//Global Defence System

var missileImage = 'https://groklearning-cdn.com/modules/4TQCjD2F82oHY2nhKbJ7L3/rocke
var collisionImage = 'https://groklearning-cdn.com/modules/wqPdjpNopABLRT8oLLYCYU/exp
var motherShipImage = 'https://groklearning-cdn.com/modules/R6YxUkGzXLKR7nYY39XhBM/mo

//missile stuff
var missile;
var missileWidth = 21;
var missileHeight = 21;
var missileFlightDirection = -1;

var vg = -20;    //exit speed of gases
var m0 = 100;    //initial mass of rocket
var deltaM = 1; //mass loss of rocket per time interval
var t = 0;       //time
var collision = false; // collision detection

var previewWidth = 1024;
var previewHeight = 740;
var finishingLineY = 600;
let gdsSpaceshipHeight = 20;
let gdsSpaceshipWidth = 20;
let gdsYSpeed = 10;
let shipsLanded = false;
let missionAccomplished = false;
var motherShip=null;

if ( typeof ySpeed.type === 'undefined')
    gdsYSpeed = 10;
else gdsYSpeed = ySpeed.type;


var el = document.createElement('p');
el.id = 'score';
el.innerHTML = '0';
document.getElementsByTagName('body')[0].appendChild(el);

el = document.createElement('p');
el.id = 'distanceToTravel';
el.innerHTML = '0';
document.getElementsByTagName('body')[0].appendChild(el);

el = document.createElement('div');
el.style.position = 'absolute';
el.style.width = '100%';
el.style.top = finishingLineY + 'px';
document.getElementsByTagName('body')[0].appendChild(el);

let line = document.createElement('hr');
el.appendChild(line);


function createMissile(left, top) {
    missile = document.createElement('img');
    missile.src = missileImage;
    missile.className = 'missile';
    missile.style.top = top + 'px';
    missile.style.left = left + 'px';
```

```javascript
        document.body.appendChild(missile);
        fadeIn(missile, 100);
        vg = -20;
        if (left < window.innerWidth / 2)
            missileFlightDirection = 1;
        else
            missileFlightDirection = -1;
        //missileFlightDirection=Math.pow(-1,getRndInteger(-1, 1));
}

function getRndInteger(min, max) {
    return Math.floor(Math.random() * (max - min + 1)) + min;
}

function moveMissile() {
    if (m0 / (m0 - t * deltaM) <= 0) {
        t = 0.95 * m0 / deltaM;
    }
    let vM = vg * Math.log(m0 / (m0 - t * deltaM)); // yes, this is indeed rocket sci
    t = t + 0.2;
    let top = parseInt(missile.style.top) + vM;
    missile.style.top = top + 'px';
    let left = parseInt(missile.style.left) + missileFlightDirection;
    missile.style.left = left + 'px';
    collisionDetection();
    determineEndGoal();

    if ((top <= -missileHeight) || (top > finishingLineY + missileHeight)) {
        document.body.removeChild(missile);
        createMissile(Math.floor(Math.random() * previewWidth), finishingLineY + miss
        collision = false;
    }
}

function collisionDetection() {
    let missileX = parseInt(missile.style.left);
    let missileY = parseInt(missile.style.top);

    let spaceships = document.querySelectorAll('.spaceship');
    let iterator = spaceships.length;
    while ((iterator > 0) && (!collision)) {
        let spaceship = spaceships[iterator - 1];
        let spaceshipX = parseInt(spaceship.style.left);
        let spaceshipY = parseInt(spaceship.style.top);
        if (!((spaceshipX + gdsSpaceshipWidth < missileX) || (missileX + missileWidth
            vg = 2 * t; //spaceship drops
            t = 0;
            missile.src = collisionImage;
            fadeOut(missile, 2000);
            document.body.removeChild(spaceship);
            document.getElementById("score").innerHTML = document.querySelectorAll('.
            collision = true;
            if (document.querySelectorAll('.spaceship').length == 0) {
                alert('Earth has defeated the Space Invaders! Don\'t give up and try
                exit();
            }
        }
        let spaceshipBottom = parseInt(spaceship.style.bottom);
        if (spaceshipBottom > finishingLineY) {
```

```javascript
            stop();
        }
        iterator--;
    }
}

function determineEndGoal() {
    let maxY = 0;
    let spaceships = document.querySelectorAll('.spaceship');
    for (let spaceship of spaceships) {
        if (parseInt(spaceship.style.top) > maxY) {
            maxY = parseInt(spaceship.style.top);
        }
    }
    let fleetSize = document.querySelectorAll('.spaceship').length;
    let message = "Fleet size: ";
    document.getElementById("score").innerHTML = message.concat(fleetSize.toString())

    let distanceToTravel = finishingLineY - maxY;
    message = "Landing in: ";
    document.getElementById("distanceToTravel").innerHTML = message.concat(distanceTo
    if (distanceToTravel <= 0) {
        shipsLanded = true;
        if (typeof xSpeed !== 'undefined')
            xSpeed = 0;
        document.body.removeChild(missile);
    }
}

function fadeIn(el, time) {
    el.style.opacity = 0;
    el.style.display = "block";

    var last = +new Date();
    var tick = function () {
        el.style.opacity = +el.style.opacity + (new Date() - last) / time;
        last = +new Date();

        if (+el.style.opacity < 1) {
            (window.requestAnimationFrame && requestAnimationFrame(tick)) || setTimeo
        }
    };

    tick();
}

function fadeOut(el, time) {
    el.style.opacity = 1;
    el.style.display = "block";

    var last = +new Date();
    var tick = function () {
        el.style.opacity = +el.style.opacity - (new Date() - last) / time;
        last = +new Date();

        if (+el.style.opacity > 0) {
            (window.requestAnimationFrame && requestAnimationFrame(tick)) || setTimeo
        }
    };
```

```javascript
    tick();
}

function newMotherShip(imageUrl) {
    motherShip = document.createElement('img');
    motherShip.src = imageUrl;
    motherShip.className = 'mothership';
    motherShip.style.left = '50%';
    motherShip.style.top = '10px';
    motherShip.style.marginLeft = '-160px';

    motherShip.style.position = 'absolute';
    document.body.appendChild(motherShip);
}

function gds() {
    if (!shipsLanded) {
        moveMissile();
        requestAnimationFrame(gds);
    }

    if (shipsLanded && !missionAccomplished) {
        if (!motherShip) {
            newMotherShip(motherShipImage);
            moveMotherShip();
        }
    }
}

createMissile(previewWidth / 2, finishingLineY + missileHeight);
gds();

function moveMotherShip() {
    let top = parseInt(motherShip.style.top);
    motherShip.style.top = top + 1 + 'px';
    if (top === 100)
        missionAccomplished = true;
    if (!missionAccomplished)
        requestAnimationFrame(moveMotherShip);
    else {
        alert('The Space Invaders have landed! One small step for a Zargan, but a gia
    }
}
function alert(text)
{
    let iDiv=document.createElement('div');
    iDiv.className='alert';
    iDiv.innerHTML=text;
    document.body.appendChild(iDiv);

    let theSpan=document.createElement('span');
    theSpan.className='closebtn';
    theSpan.setAttribute('onclick', 'this.parentElement.style.display=\'none\';');
    theSpan.innerHTML='&times';
    iDiv.appendChild(theSpan);
}
```

stylesheet.css

stylesheet.css

```css
.spaceship {
    position: absolute;
}
.missile {
    position: absolute;
}
canvas {
    border: 1px dotted white;
    background: white;
}

body {
    background: black;
    background-image: url('https://groklearning-cdn.com/modules/pwgzeGenotee9mfvT6Qmt
    background-repeat: no-repeat;
    background-attachment: fixed;
    background-position: center;
}

p {
    color: red;
    font-size: 20px;
}

.mothership {
    position: relative;
    display: inline-block;
    margin: 50px;
    border-radius: 25%;
    background-color: #0ff;
    box-shadow: 0 0 40px 20px #fff, 0 0 100px 50px #0ff;
}

.alert {
    padding:10px;
    background-color: rgba(255,0,0,0.8);
    color: white;
    position:absolute;
    left : 50%;
    top : 1%;
    width:300px;
    z-index:1000;
    border-radius: 10px;
    margin-left:-161px;
}

.closebtn {
    margin-left: 15px;
    color: white;
    font-weight: bold;
    float: right;
    font-size: 22px;
    line-height: 20px;
    cursor: pointer;
    transition: 0.3s;
}

.closebtn:hover {
    color: black;
```

```
}
```

Testing

☐ Marking is not available for the Playground.