



Australian
Computing
Academy

DT Challenge JavaScript
Space Invaders

1. The journey begins
2. Growing the fleet
3. Stardrive operation
4. The battle for Earth



[\(https://creativecommons.org/licenses/by/4.0/\)](https://creativecommons.org/licenses/by/4.0/)

The Australian Digital Technologies Challenges is an initiative of, and funded by the [Australian Government Department of Education and Training](https://www.education.gov.au/) (<https://www.education.gov.au/>).

© Australian Government Department of Education and Training.

1

THE JOURNEY BEGINS

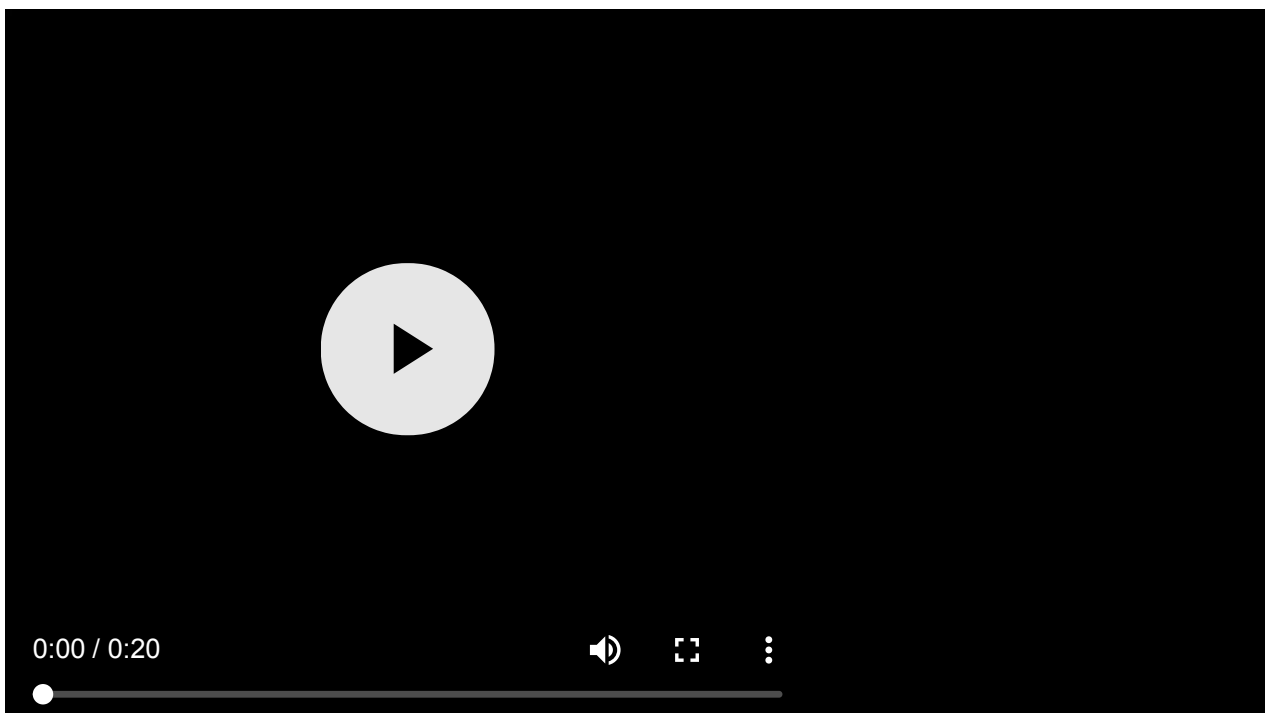
1.1. The journey begins

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.



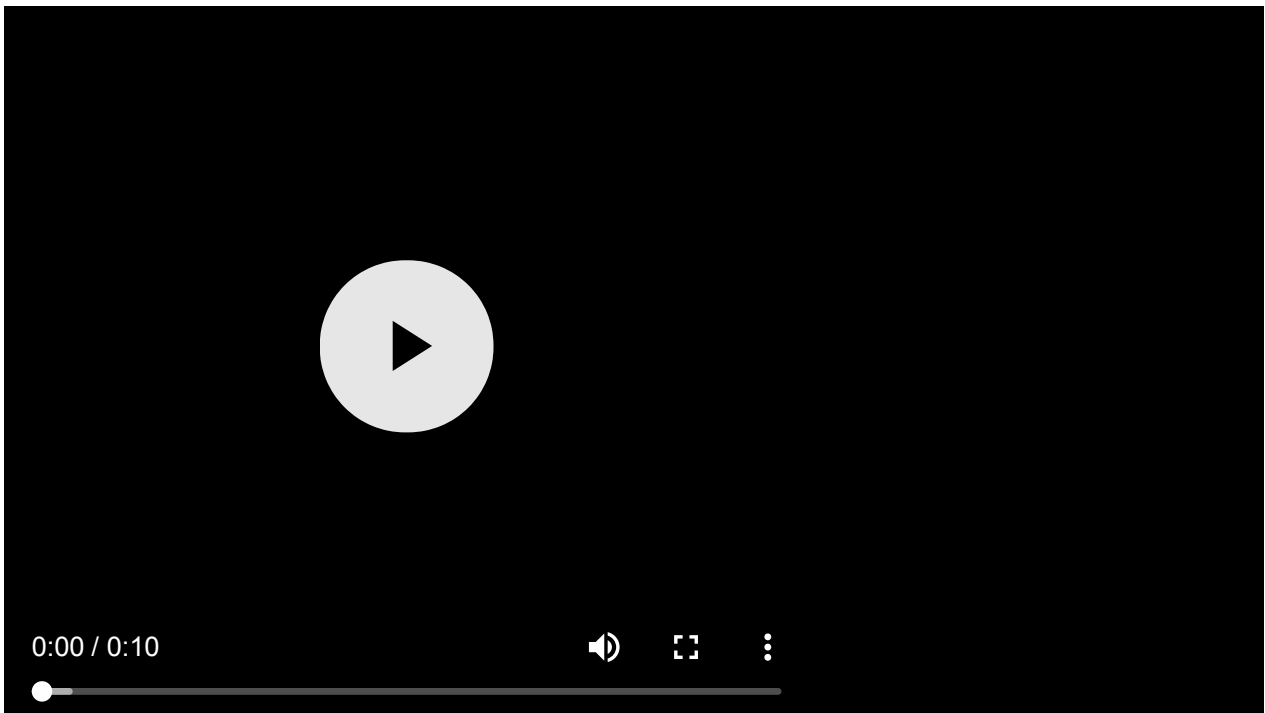
Planet Zarg_(https://groklearning-cdn.com/modules/Y6xAC38rZ6JenitbkLfrmd/planet_zarg480p.mov).

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.

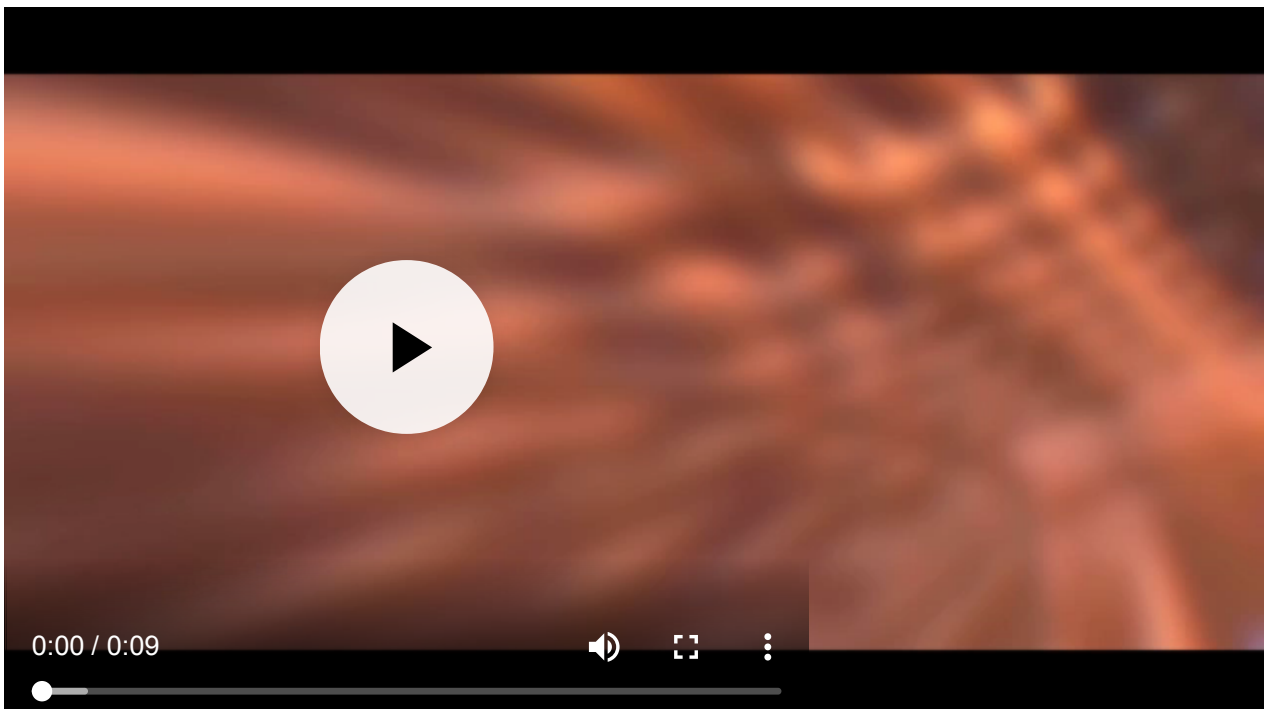
Be advised, the United Nations of Earth has a powerful defence system.



Earth (<https://groklearning-cdn.com/modules/Z2fp8ziBcAYrLuedYEomdN/securewater480p.mov>).

1.1.3. Space station

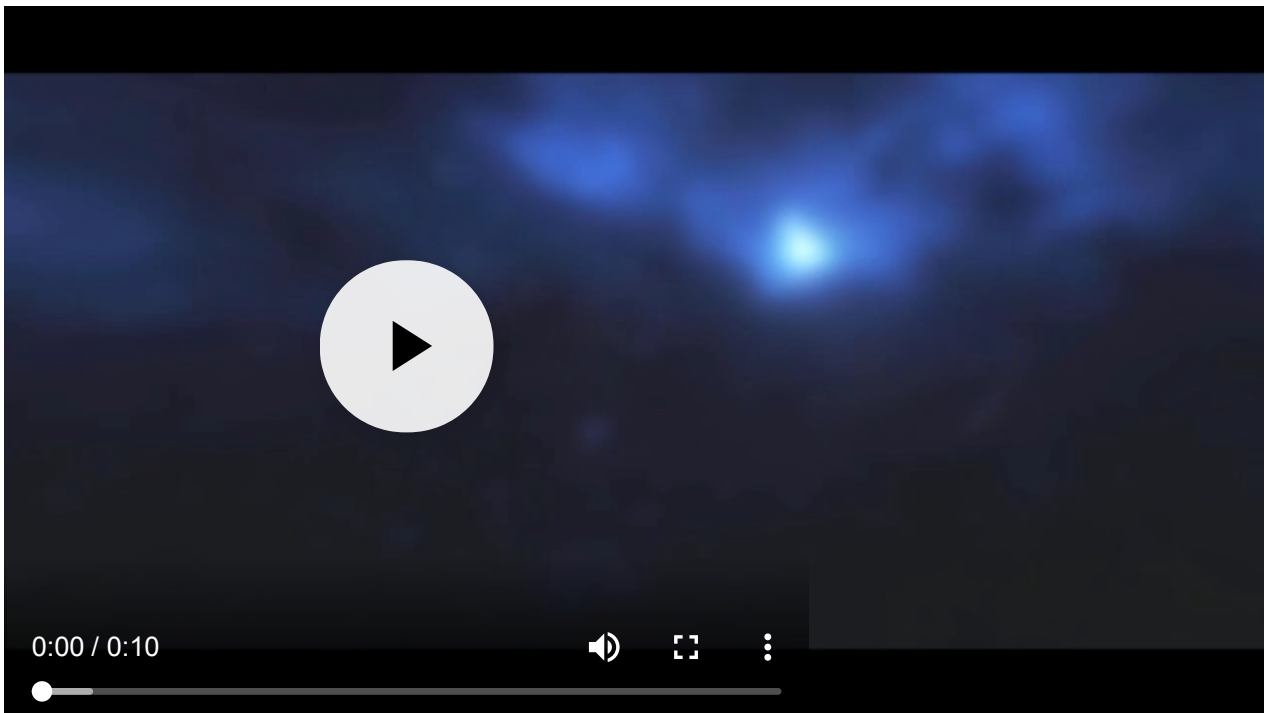
Commander, for the past five years, Zarg's best engineers have constructed a space station in orbit. There, we are building the trans-warp spaceship that will take you to Earth safely.



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 \(https://groklearning-cdn.com/modules/BaNrDDHE8wz8jiAe9EfrCn/spaceship_leaving_zarg.mov\).](https://groklearning-cdn.com/modules/BaNrDDHE8wz8jiAe9EfrCn/spaceship_leaving_zarg.mov)

1.1.5. The end goal

Commander, we will be 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 secure Earth's abundant water resources.



[_ \(https://groklearning-cdn.com/modules/kPDtAbVNRtj9kuCSoTdwa8/endgoal.png\)](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 **program.js** file) 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  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 them wisely.

Do each of the following steps in the computer command interface on the right.

If you get stuck, take a peek into the yellow box.

Follow the steps

[Step 1: Create an image to represent the spaceship](#)

```
var spaceship = document.createElement('img');
```

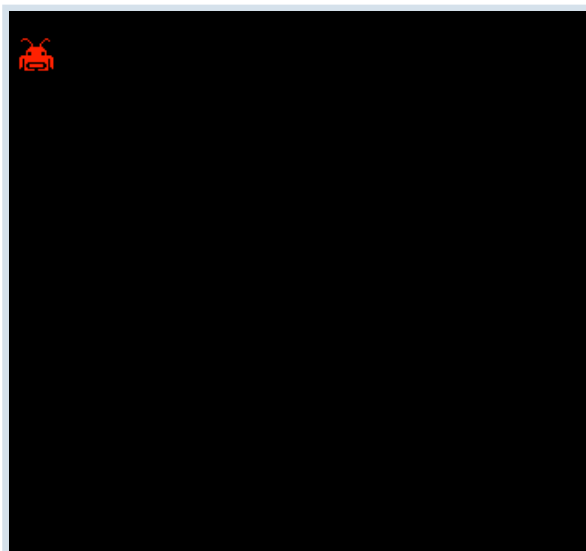
- We use an HTML *image element* to represent our spaceship;
- `document.createElement('img')` creates the element;
- The `spaceship` *variable* holds the element for later use.

[Step 2: Choose an image for the spaceship](#)

[Step 3: Set the spaceship's class name](#)

[Step 4: Insert your spaceship into the simulation](#)

This is the result



[_ \(https://groklearning-](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.

Hint - Click here to see the complete code if you are stuck

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 target. They want you to test it.

Change Spaceship Design

To change the spaceship design, change the string from: `'spaceship1.png'` into `'spaceship2.png'`

This is the result:



[.\(https://groklearning-](https://groklearning-)

[cdn.com/problems/Uq8eUrHKDx73VTCW8puDR2/spaceship3-screenshot.png\)](https://cdn.com/problems/Uq8eUrHKDx73VTCW8puDR2/spaceship3-screenshot.png)

Spaceship2 Design

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

You'll need

`program.js`

```
var spaceship = document.createElement('img');
spaceship.src = 'spaceship1.png';
spaceship.className = 'spaceship';
document.body.appendChild(spaceship);
```

Testing

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



1.1.9. Problem: More spaceship designs

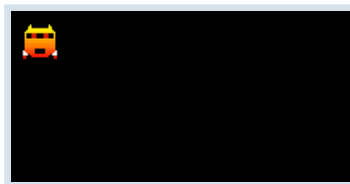
Commander, the science department has developed more cool-looking spaceship designs. You now have access to additional top secret spaceship looks.

Use the code from the previous mission.

Change Spaceship Design

- To change the spaceship design, you will only need to adjust the following line
- `spaceship.src = 'spaceship2.png';`
- You can change `'spaceship2.png'` to any of the following:
 - `'spaceship3.png'`
 - `'spaceship4.png'`
 - `'spaceship5.png'`
 - `'spaceship6.png'`
 - `'spaceship7.png'`
 - `'spaceship8.png'`

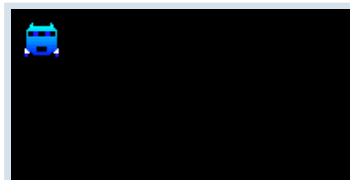
Try out all the new looks:



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

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

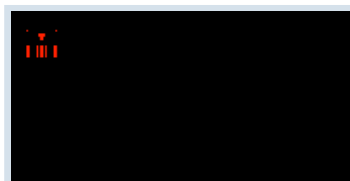
spaceship3 - Inferno



([https://groklearning-](https://groklearning-cdn.com/problems/zqBhB5puN9GgF2irPVBaZ7/spaceship5-screenshot.png)

[cdn.com/problems/zqBhB5puN9GgF2irPVBaZ7/spaceship5-screenshot.png\)](https://groklearning-cdn.com/problems/zqBhB5puN9GgF2irPVBaZ7/spaceship5-screenshot.png)

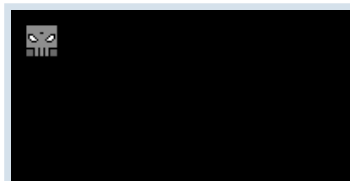
spaceship4 - Blizzard



([https://groklearning-](https://groklearning-cdn.com/problems/F3yq6LGLaa2y5odjDrFbLX/spaceship6-screenshot.png)

[cdn.com/problems/F3yq6LGLaa2y5odjDrFbLX/spaceship6-screenshot.png\)](https://groklearning-cdn.com/problems/F3yq6LGLaa2y5odjDrFbLX/spaceship6-screenshot.png)

spaceship5 - Stealth



([https://groklearning-](https://groklearning-cdn.com/problems/p3RDwkg4UdCKkiw2veW6MH/spaceship7-screenshot.png)

[cdn.com/problems/p3RDwkg4UdCKkiw2veW6MH/spaceship7-screenshot.png\)](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.

Which spaceship design do you like most? 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.

You'll need

 `program.js`

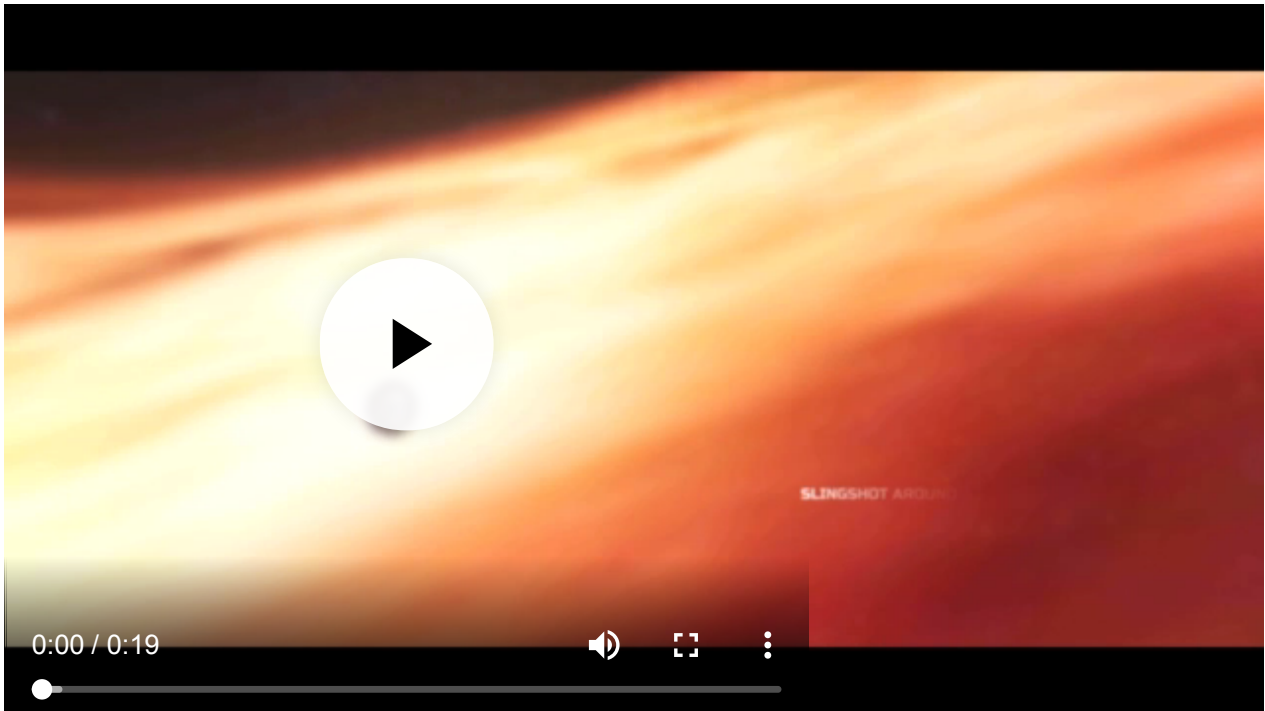
```
var spaceship = document.createElement('img');
spaceship.src = 'spaceship2.png';
spaceship.className = 'spaceship';
document.body.appendChild(spaceship);
```

Testing

☐ Testing the existence of the first spaceship. Good job. Your code has passed our test.

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.



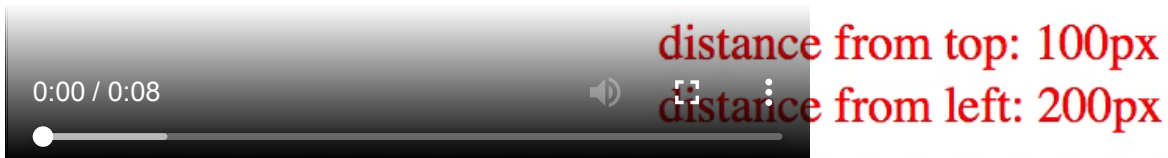
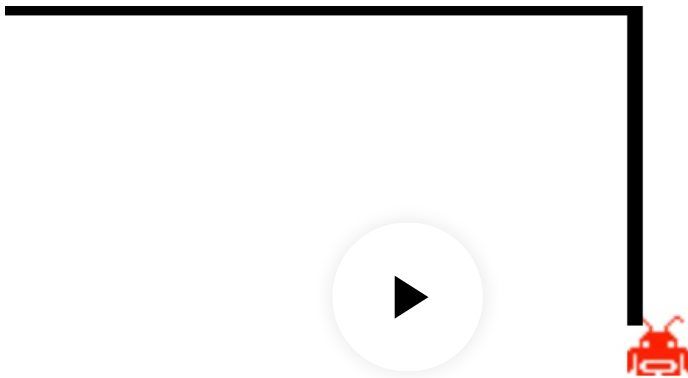
[Black Hole \(https://groklearning-cdn.com/modules/JEovCdw49Sd9BQPrhUXBzb/black_hole.mov\)](https://groklearning-cdn.com/modules/JEovCdw49Sd9BQPrhUXBzb/black_hole.mov).

1.1.11. Positioning spaceships

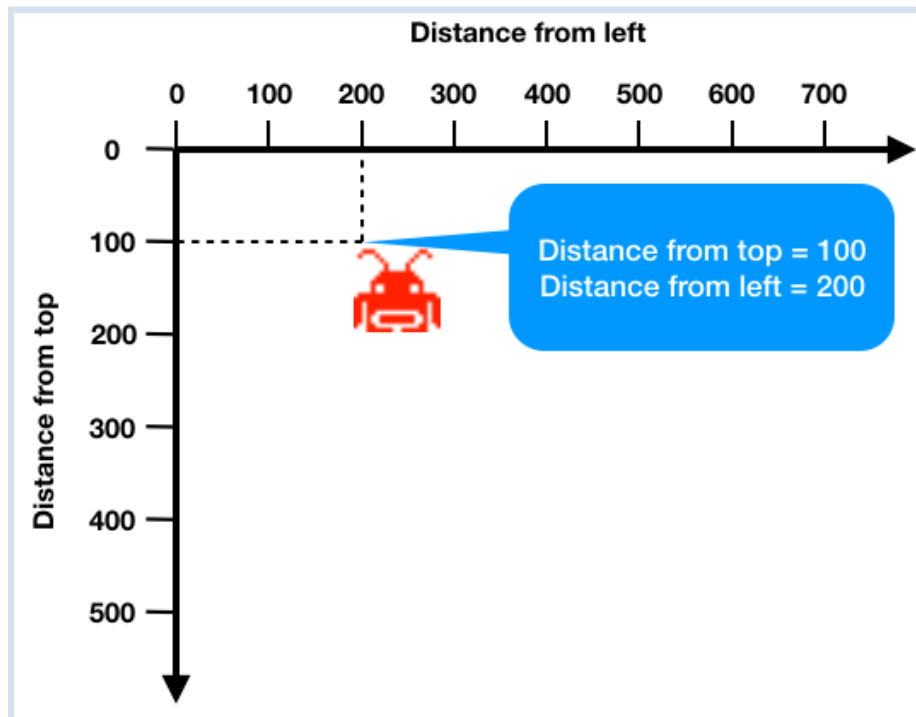
Commander, let's work out how to position spaceships. Our maths geniuses have found an elegant way to do this.

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 100 pixels from the top and 200 pixels from the left.



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



([https://groklearning-](https://groklearning-cdn.com/modules/DhWwLw58xjNweyYWm7jHfb/cartesiancoordinatesystem.png)

[cdn.com/modules/DhWwLw58xjNweyYWm7jHfb/cartesiancoordinatesystem.png](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.

Use the code from the previous mission.

Position Spaceship

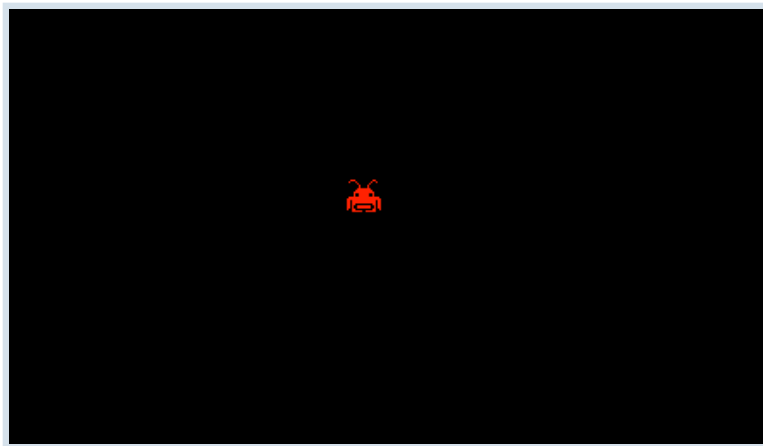
Set the spaceship's position from the top:

```
spaceship.style.top= 100+ 'px';
```

Set the spaceship's position from the left:

```
spaceship.style.left=200+ 'px';
```

The result will look like this:



[_ \(https://groklearning-cdn.com/problems/KEzcB9yznnJ3vDKPFtrSK/si-step1.png\)](https://groklearning-cdn.com/problems/KEzcB9yznnJ3vDKPFtrSK/si-step1.png)

A spaceship

💡 Hint - Click here to see the complete code

Congratulations, Commander. You now know how to position your spaceships freely.

We will practice this in the following mission.

You'll need

`program.js`

```
var spaceship = document.createElement('img');
spaceship.src = 'spaceship1.png';
spaceship.className = 'spaceship';
document.body.appendChild(spaceship);
```

Testing

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

1.1.13. Problem: Positioning a spaceship: manoeuvre



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

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

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.js](#)

```
var spaceship = document.createElement('img');
spaceship.src = 'spaceship1.png';
document.body.appendChild(spaceship);
spaceship.className = 'spaceship';
spaceship.style.top= 100+'px';
spaceship.style.left=200+'px';
```

Testing

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

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 ?

☐ spaceship.style.left = 10+'px'; spaceship.style.top = 50+'px'

☐ spaceship.style.left = 10; spaceship.style.top = 50;

☐ spaceship.style.left = 10+'px'; spaceship.style.top = 50+'px';

☐ spaceship.style.left = 100+'px'; spaceship.style.top = 50+'px';

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.

Long live the Zargan Empire.



Captain's Badge

2

GROWING THE FLEET

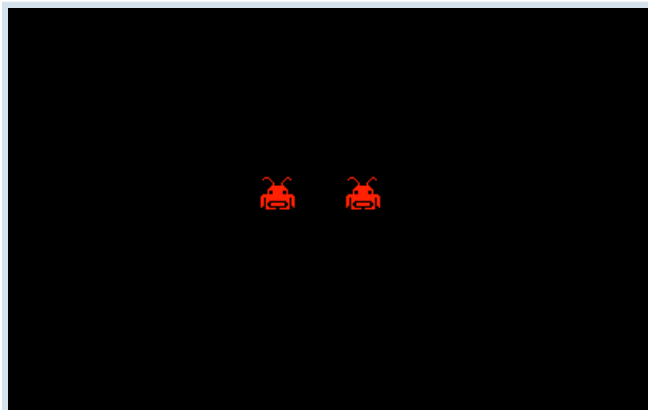
2.1. Growing the fleet

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.



[_ \(https://groklearning-cdn.com/modules/373gJwL7YMZPEszktefWsc/si-step2.png\)](https://groklearning-cdn.com/modules/373gJwL7YMZPEszktefWsc/si-step2.png)

Two Spaceships

2.1.2. Problem: Making two spaceships



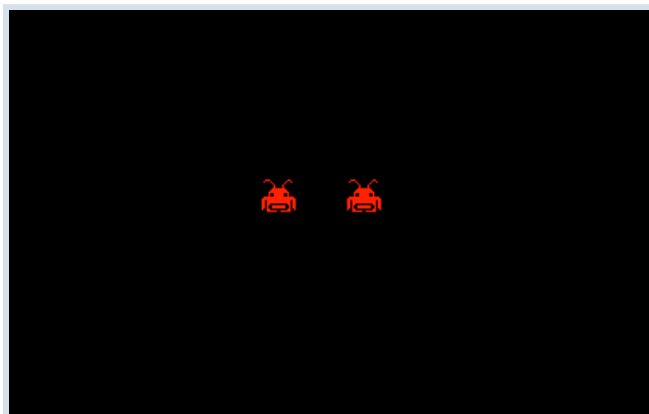
Captain, you are now going to build two spaceships.

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

💡 Building two Spaceships

- You will need to create a second spaceship object. Call it spaceship2.

The result should look something like this:



cdn.com/modules/373gJwL7YMZPEszktefWsc/si-step2.png

Two Spaceships

💡 Hint - Click here to see the complete code

Congratulations, Captain. You have successfully made two spaceships.

You'll need

[program.js](#)

```
var spaceship = document.createElement('img');
spaceship.src = 'spaceship1.png';
document.body.appendChild(spaceship);
spaceship.style.top = 100+'px';
spaceship.style.left = 150+'px';
spaceship.className = 'spaceship';

var spaceship2 = document.createElement('img');
```

Testing

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

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 write 6 additional lines of code:

```
var spaceship = document.createElement('img');
spaceship.src = spaceshipImage;
spaceship.style.top= 100+'px';
spaceship.style.left=150+'px';
spaceship.className = 'spaceship';
document.body.appendChild(spaceship);
```

They fear that making spaceships this way will slow down the invasion of Earth. For 100 spaceships, we would need $100 \times 6 = 600$ lines of code. 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.

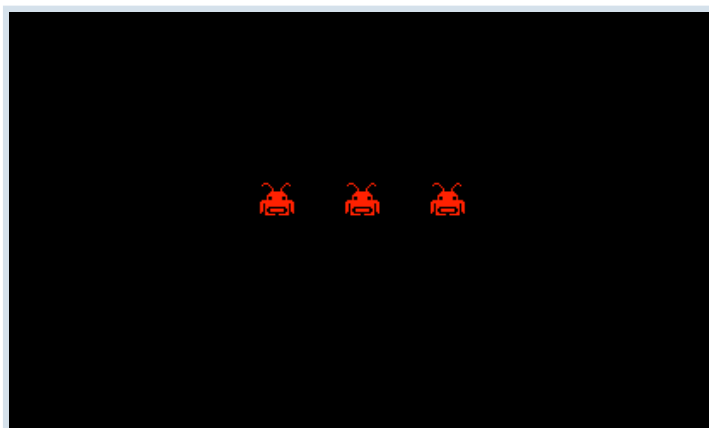
To use it, place the code that makes one spaceship into a function, called `createSpaceship`

```
function createSpaceship(left, top) {
  let spaceship = document.createElement('img');
  spaceship.src = 'spaceship1.png';
  document.body.appendChild(spaceship);
  spaceship.className = 'spaceship';
  spaceship.style.top = top + 'px';
  spaceship.style.left = left + 'px';
}
```

Then, you can call the function as often as you need. Let's now make three spaceships in a row, 100 pixels from the top and 150, 200, and 250 pixels from the left.

```
createSpaceship(150,100);
createSpaceship(200,100);
createSpaceship(250,100);
```

The result should look something like this:



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

Three Spaceships

[_ \(https://groklearning-](https://groklearning-)

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 600. The invasion is on track again.

Note that we use `let` to declare the spaceship variable (rather than `var`) to restrict it to the function. The spaceship variable is only visible inside the function. This is a clever way to avoid that some of our other code that we will write later unintentionally makes changes to the spaceships.

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

You'll need

 `program.js`

```
function createSpaceship(left, top) {
```

Testing

- ☐ Testing the position of three spaceships. Good job. Your code has 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/pAiKyyAJ5VC4aZR3pgELBR/si-step4.png)

A Row of Spaceships

You'll need

`program.js`

```
function createSpaceship(left, top) {
  let spaceship = document.createElement('img');
  spaceship.src = 'spaceship1.png';
  document.body.appendChild(spaceship);
  spaceship.className = 'spaceship';
  spaceship.style.top = top + 'px';
  spaceship.style.left = left + 'px';
}

createSpaceship(30,50);
createSpaceship(60,50);
createSpaceship(90,50);
```

Testing

- ☐ Testing the position of ten spaceships. Good job. Your code has passed our test.

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 linked in 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 code already in the window on the right.

```
for (let spaceshipCounter = 0; spaceshipCounter < 10; spaceshipCounter++){
    createSpaceship(startLeft, 50);
    startLeft=startLeft+20+10;
}
```

>the main idea is that we call the '**createSpaceship**' function ten 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:



cdn.com/problems/pAiKyyAJ5VC4aZR3pqELBR/si-step4.png

A Row of Spaceships

[_ \(https://groklearning-](https://groklearning-)

💡 Hint - Click here to see the complete code

You'll need

[program.js](#)

```
var startLeft = 30;
function createSpaceship(left, top) {
    let spaceship = document.createElement('img');
    spaceship.src = 'spaceship1.png';
    document.body.appendChild(spaceship);
    spaceship.className = 'spaceship';
    spaceship.style.top = top + 'px';
    spaceship.style.left = left + 'px';
}
```

Testing

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

☐ Testing the position of ten spaceships. Good job. Your code has passed our test.

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 start at 30 px from the left.

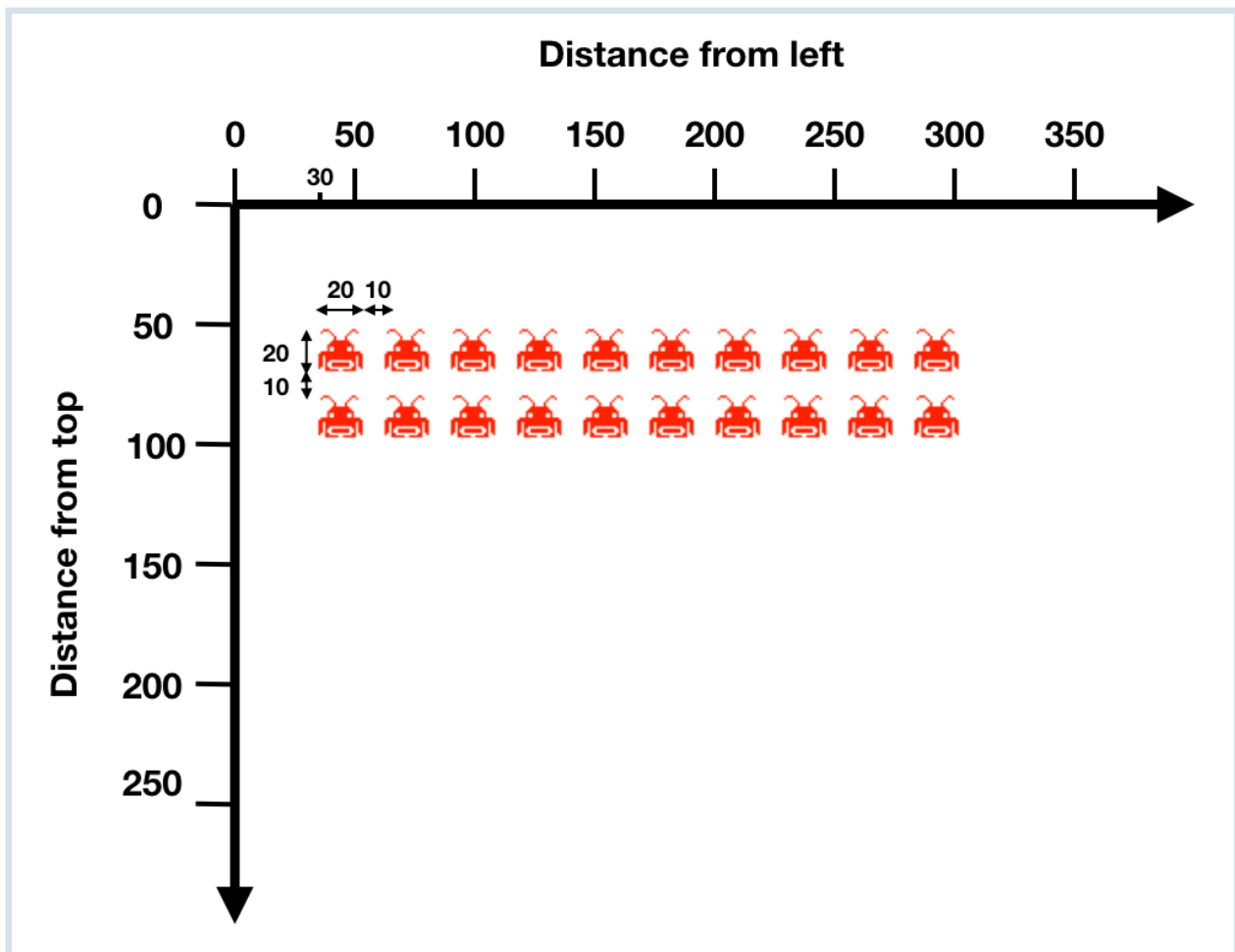
💡 Hints

Use the code from the previous mission.

A spaceship is 20 pixels tall.

You will need a second for-loop.

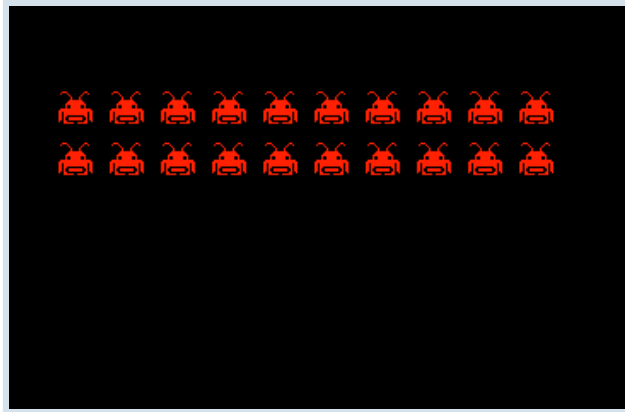
Here is a schematic drawing with the measurements you need.



(<https://groklearning-cdn.com/problems/nB8AnF6nAQ6RvkKKF77aFA/st-step4-22rowsofspaceships.png>)

Two rows of Spaceships

The result should look something like this:



[_ \(https://groklearning-](https://groklearning-)

cdn.com/problems/CvqWdQhfwjMT3UajGebGYM/si-step4-2endresult.png

Two rows of Spaceships

💡 Hint - Click here to see the complete code

You'll need

program.js

```
var startLeft = 30;

function createSpaceship(left, top) {
  let spaceship=document.createElement('img');
  spaceship.src = 'spaceship1.png';
  document.body.appendChild(spaceship);
  spaceship.className = 'spaceship';
  spaceship.style.top = top + 'px';
  spaceship.style.left = left + 'px';
}
for (let spaceshipCounter = 0; spaceshipCounter < 10; spaceshipCounter++){
  createSpaceship(startLeft, 50);
  startLeft=startLeft+20+10;
}
```

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 our code and have some suggestions.

They say that the loops only differ a tiny little bit by the value of the distance from the top:

```
createSpaceship(startLeft, 50);  
...  
createSpaceship(startLeft, 80);
```

They say that if we wanted to make many rows of spaceships we could put that whole loop-code into a new function and call it as often as we need to.

They say that this is similar to before when we tried to make several spaceships.

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

2.1.10. Problem: Three rows of spaceships



Captain, we will now make three rows of spaceships. We re-use `createSpaceship(left, top)` and place the for-loop from our previous experiment into a new function that we call `createRow(left, top)`.

💡 Hints

We also add a couple more variables, such as the gap between spaceships, and positions, so that we don't have to repeat them in our code.

```
var gap = 5; // the gap in pixels between two spaceships
var spaceshipsPerRow = 2; // the number of spaceships in one row
var startLeft = 0; // the left starting point of the first spaceship
var spaceshipWidth=20; // the width of a spaceship
```

We can then call `createRow(left, top)` as often as we need, for example three times to make three rows of spaceships.

```
createRow(startLeft, 50);
createRow(startLeft, 80);
createRow(startLeft, 110);
```

Change the content of some of the variables in the editor to make three rows of 10 spaceships that are 10 pixels apart (gap). The first spaceship in each row should be positioned 30 pixels from the left

The result should look something like this:



[_\(https://groklearning-](https://groklearning-)

[cdn.com/problems/kCWZLJGe9hNTbF5DAq3ZnJ/si-step4-3threerows.png\)](https://groklearning-cdn.com/problems/kCWZLJGe9hNTbF5DAq3ZnJ/si-step4-3threerows.png)

Three rows of spaceships

You'll need

program.js

```
var gap = 5; // the gap in pixels between two spaceships
var spaceshipsPerRow = 2; // the number of spaceships in one row
var startLeft = 0; // the left starting point of the first spaceship
var spaceshipWidth=20; // the width of a spaceship

function createSpaceship(left, top) {
    let spaceship =document.createElement('img');
    spaceship.src = 'spaceship1.png';
    document.body.appendChild(spaceship);
    spaceship.className = 'spaceship';
    spaceship.style.top = top + 'px';
    spaceship.style.left = left + 'px';
}
function createRow(left, top){
    for (let spaceshipCounter = 0; spaceshipCounter < spaceshipsPerRow; spaceshipCounter++) {
        createSpaceship(left, top);
        left=left+spaceshipWidth+gap;
    }
}
createRow(startLeft, 50);
createRow(startLeft, 80);
createRow(startLeft, 110);
```

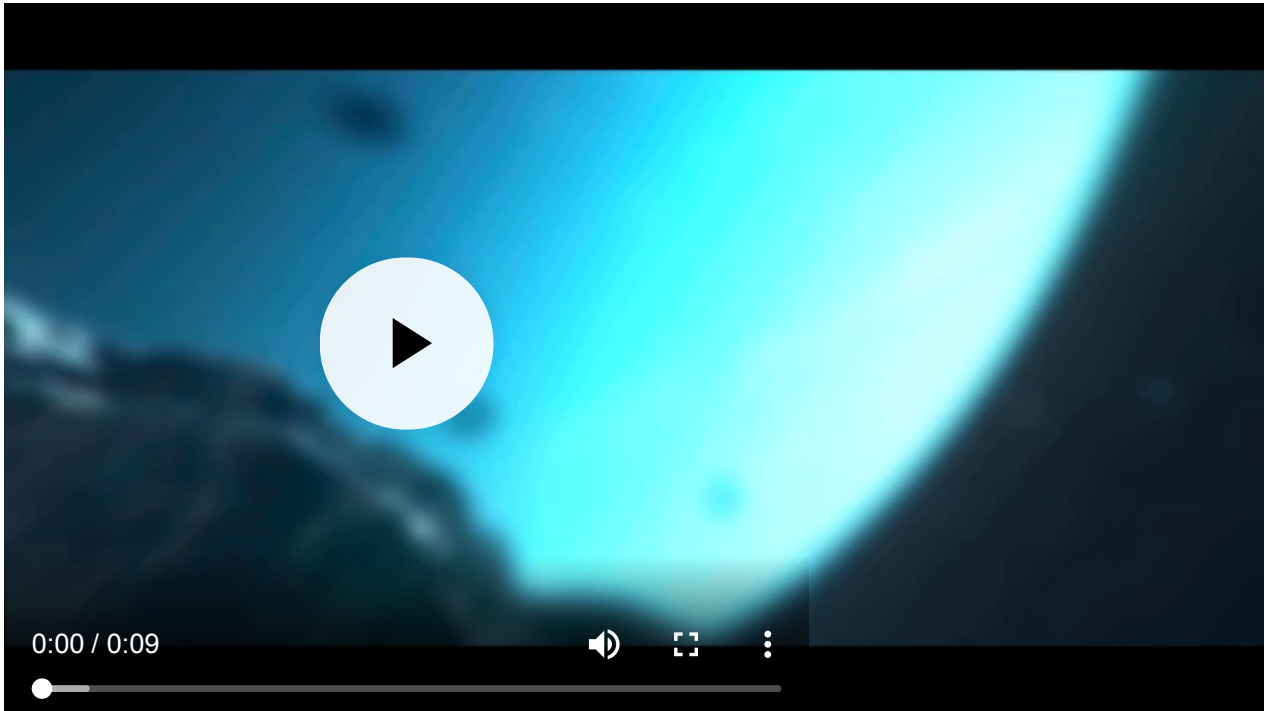
Testing

- ☐ 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 half way station on our journey to Earth.

Here, we will mine for precious minerals so that our 3D spaceship printer can make more spaceships.



[The Moon of Xobron IV \(https://groklearning-cdn.com/modules/hSf9s4kEwcCGmAGcAWdiW3/XobronIV480p.mov\)](https://groklearning-cdn.com/modules/hSf9s4kEwcCGmAGcAWdiW3/XobronIV480p.mov).

2.1.12. Problem: Grid of spaceships



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

THE DOCTOR has suggested to place the code that calls the `createRow(left, top)` function inside another loop and place this into a new function `createGrid()`

💡 Hints

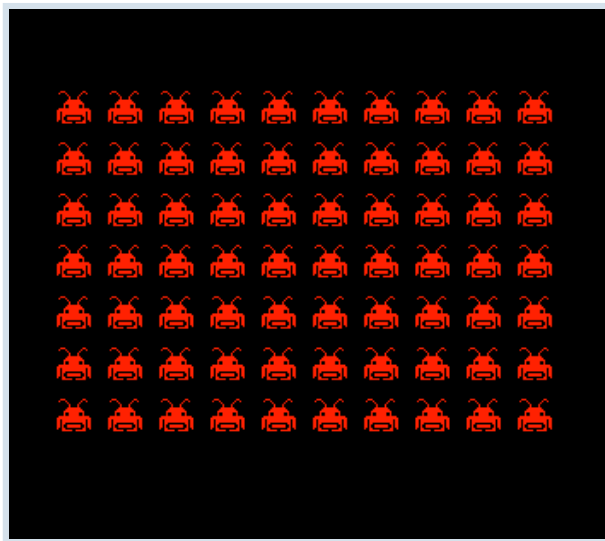
We make three more variables. One to set the number of rows of spaceships we want to build, and another that sets the distance from the top to the first row of spaceships.

```
var rowsOfSpaceships = 7;
var startTop = 50;
var spaceshipHeight=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 the new function.

- Make 7 rows of spaceships
- The first row should start 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:



cdn.com/problems/ExhjWwLKGfaYk8pWqFN9yj/si-step5.png

A seven-row grid of spaceships

[_ \(https://groklearning-](https://groklearning-)

💡 Hint - Click here to see the complete code

You'll need

program.js

```
var gap = 10;
var spaceshipsPerRow = 10;
var startLeft = 30;
var spaceshipWidth=20;

function createSpaceship(left, top) {
    let spaceship =document.createElement('img');
    spaceship.src = 'spaceship1.png';
    document.body.appendChild(spaceship);
    spaceship.className = 'spaceship';
    spaceship.style.top = top + 'px';
    spaceship.style.left = left + 'px';
}
function createRow(left, top){
    for (let spaceshipCounter = 0; spaceshipCounter < spaceshipsPerRow; spaceshipCoun
        createSpaceship(left, top);
        left=left+spaceshipWidth+gap;
    }
}
function createGrid() {
    //add the code that creates a grid of spaceships here
}

//then, don't forget to call createGrid()
```

Testing

- ☐ Testing the position of seven rows of spaceships. Good job. Your code has passed our test.

2.1.13. Problem: Grid of spaceships quiz



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

- ☐ 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(left, top), which in turn calls createSpaceship makes spaceships

2.1.14. Commodore

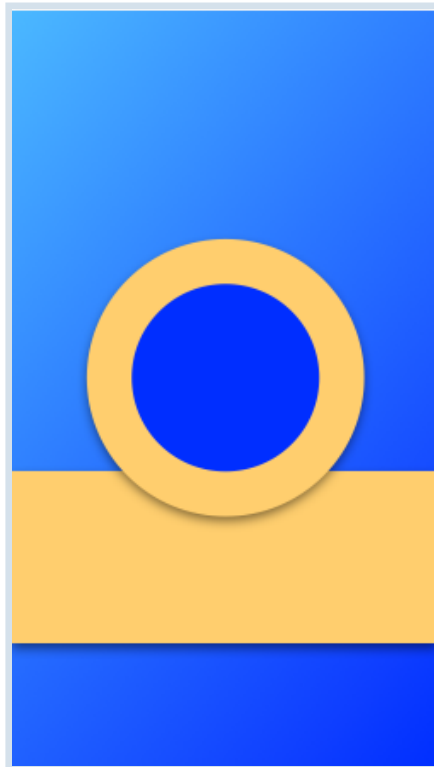
Captain, congratulations on the successful completion of the fleet building program. The Zargan High Command is very proud of your achievements.

The High Command hereby promotes you to the rank of:

Commodore

The people of planet Zarg congratulate. 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

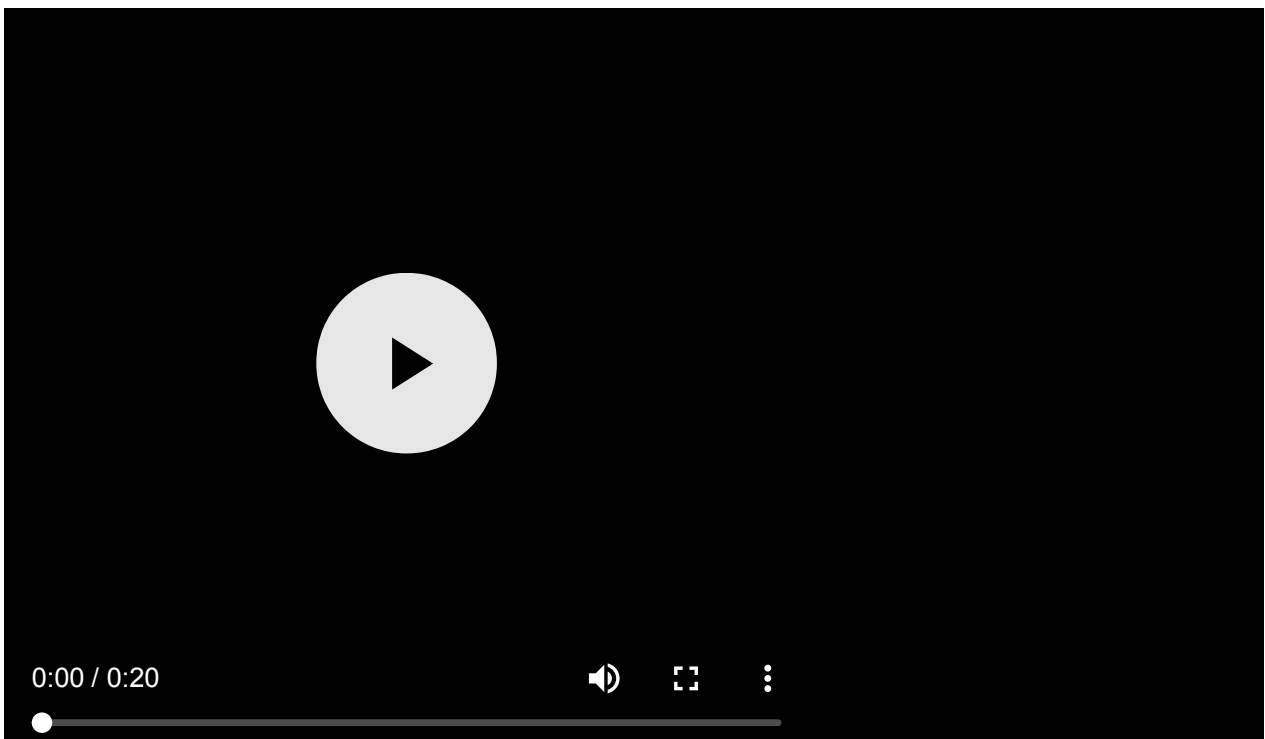
Commodore, you have successfully created the Zargan space fleet.

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:



[The Zargan Fleet: Attack Formation \(https://groklearning-cdn.com/modules/icgk9aSDc69dzt4wzBGLgS/missionbriefing3sound.mp4\)](https://groklearning-cdn.com/modules/icgk9aSDc69dzt4wzBGLgS/missionbriefing3sound.mp4).

3.1.2. Zeti Beta II

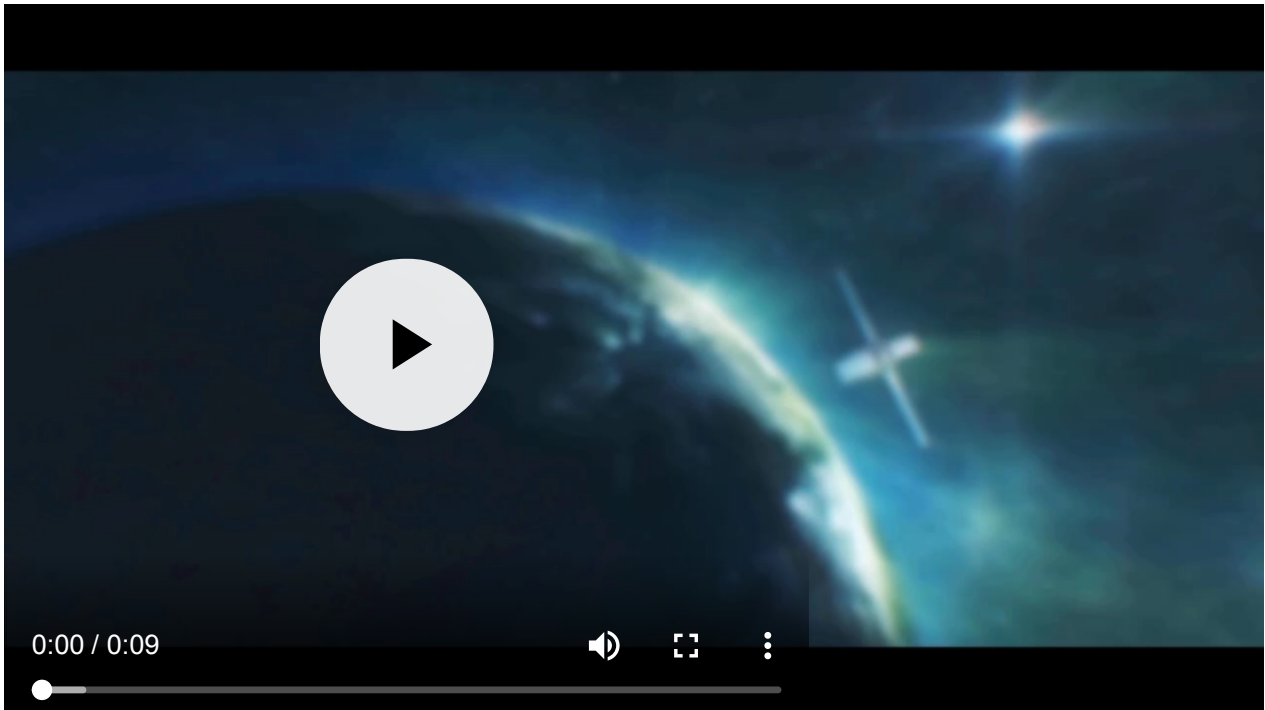
Commodore, we will conduct our final preparations in the shelter of Zeti Beta II.

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

Two years ago, the High Command has deployed a spy satellite into orbit around Zeti Beta II to check if maybe the earthlings know this planet.

Luckily for us, they don't.

Helm reports that we are about to enter into orbit.



[Zeti Beta II \(https://groklearning-cdn.com/modules/5BgdhJM5CsQZhFRhr2tzi3/zeti-beta_ii-480p.mov\)](https://groklearning-cdn.com/modules/5BgdhJM5CsQZhFRhr2tzi3/zeti-beta_ii-480p.mov).

3.1.3. Problem: Moving spaceships right



Commodore, we first move spaceships to the right. Do do this, you need to update the spaceships' distance from the left repeatedly by just 1 pixel.

- Make 5 rows of 10 spaceships.
- The first row should start 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

These are the steps:

[Step 1: Make a new function moveSpaceships](#)

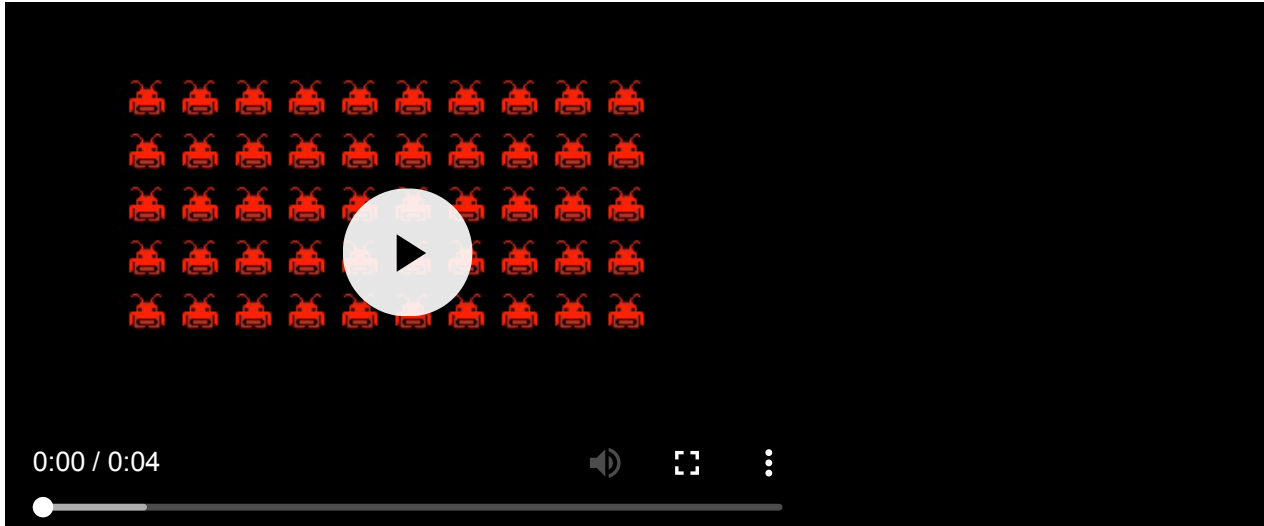
[Step 2: Loop across all spaceships](#)

[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 \(https://groklearning-cdn.com/problems/jaUoPAEnmzv7ciD6cBABEk/spaceships-moving-right.mov\).](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.js

```
var gap = 10;
var spaceshipsPerRow = 10;
var startLeft = 30;
var startTop = 50;
var spaceshipWidth = 20;
var spaceshipHeight = 20;
var rowsOfSpaceships = 5;

function createSpaceship(left, top) {
    let spaceship = document.createElement('img');
    spaceship.src = 'spaceship1.png';
    document.body.appendChild(spaceship);
    spaceship.className = 'spaceship';
    spaceship.style.top = top + 'px';
    spaceship.style.left = left + 'px';
}

function createRow(left, top) {
    for (let spaceshipCounter = 0; spaceshipCounter < spaceshipsPerRow; spaceshipCounter++) {
        createSpaceship(left, top);
        left = left + spaceshipWidth + gap;
    }
}

function createGrid() {
    let top = startTop;
    for (let row = 0; row < rowsOfSpaceships; row++) {
        createRow(startLeft, top);
        top = top + spaceshipHeight + gap;
    }
}

// add the new function moveSpaceships here
createGrid();
moveSpaceships();
```

Testing

- ☐ Testing the animation of the fleet. Good job. Your code has passed our test.

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



Commodore, we want 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: Make a new function](#)

[Step 4a: Make a decision](#)

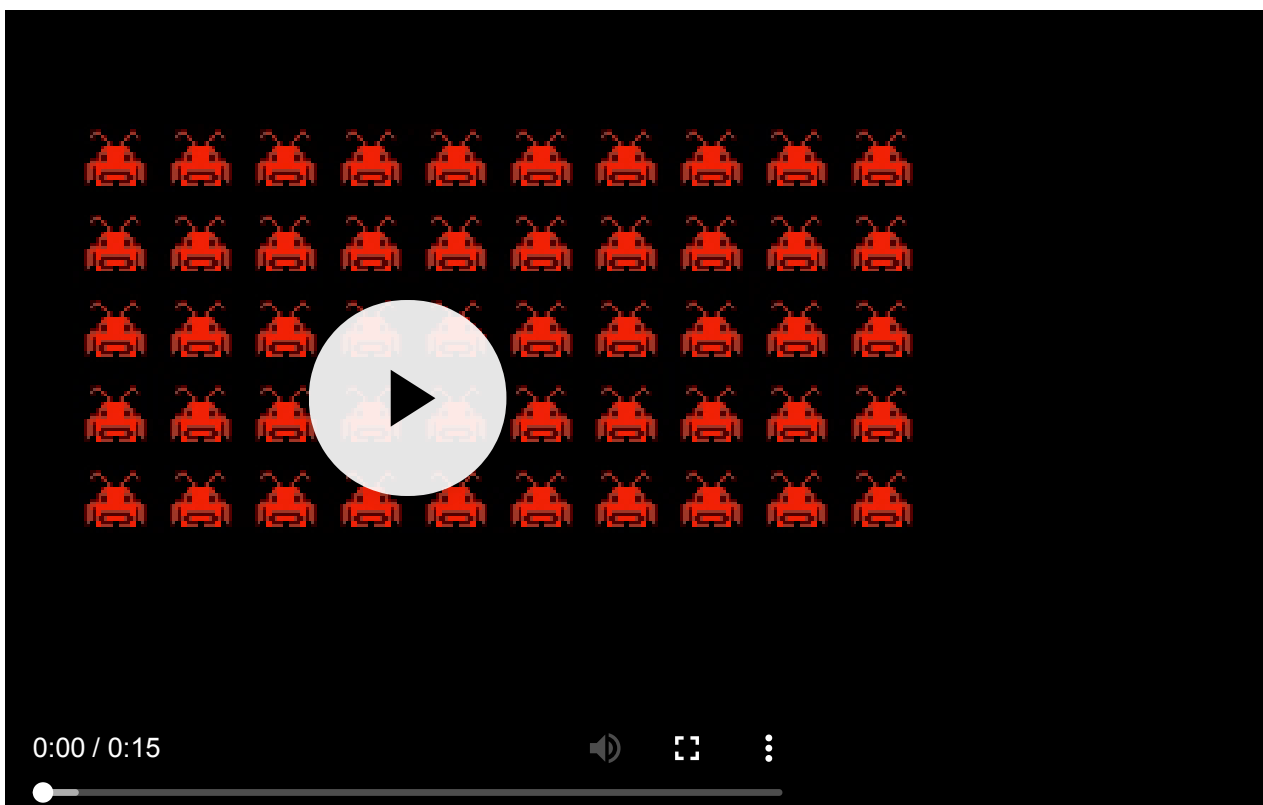
[Step 4b: Call moveSpaceships\(\)](#)

[Step 4c: Update xDistance](#)

[Step 4d: Animate](#)

[Step 5: Call the new function](#)

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.js`

```
var gap = 10;
var spaceshipsPerRow = 10;
var startLeft = 30;
var startTop = 50;
var spaceshipWidth = 20;
var spaceshipHeight = 20;
var rowsOfSpaceships = 5;

function createSpaceship(left, top) {
    let spaceship = document.createElement('img');
    spaceship.src = 'spaceship1.png';
    document.body.appendChild(spaceship);
    spaceship.className = 'spaceship';
    spaceship.style.top = top + 'px';
    spaceship.style.left = left + 'px';
}

function createRow(left, top) {
    for (let spaceshipCounter = 0; spaceshipCounter < spaceshipsPerRow; spaceshipCounter++) {
        createSpaceship(left, top);
        left = left + spaceshipWidth + gap;
    }
}

function createGrid() {
    let top = startTop;
    for (let row = 0; row < rowsOfSpaceships; row++) {
        createRow(startLeft, top);
        top = top + spaceshipHeight + gap;
    }
}

function moveSpaceships() {
    let spaceships = document.querySelectorAll('.spaceship');
    for (let spaceship of spaceships) {
        let left = parseInt(spaceship.style.left) + 1;
        spaceship.style.left = left + 'px';
    }
}

createGrid();
```

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:



[Spaceships moving right-left-right-... \(https://groklearning-cdn.com/problems/3mfubX87hcZMqnirDLvWfZ/spaceships-oscillating.mov\).](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.js

```

var gap = 10;
var spaceshipsPerRow = 10;
var startLeft = 30;
var startTop = 50;
var spaceshipWidth = 20;
var spaceshipHeight = 20;
var rowsOfSpaceships = 5;
var xSpeed = 1;
var xDistance = 1;

function createSpaceship(left, top) {
    let spaceship = document.createElement('img');
    spaceship.src = 'spaceship1.png';
    document.body.appendChild(spaceship);
    spaceship.className = 'spaceship';
    spaceship.style.top = top + 'px';
    spaceship.style.left = left + 'px';
}

function createRow(left, top) {
    for (let spaceshipCounter = 0; spaceshipCounter < spaceshipsPerRow; spaceshipCounter++) {
        createSpaceship(left, top);
        left = left + spaceshipWidth + gap;
    }
}

function createGrid() {
    let top = startTop;
    for (let row = 0; row < rowsOfSpaceships; row++) {
        createRow(startLeft, top);
        top = top + spaceshipHeight + gap;
    }
}

function moveSpaceships() {
    let spaceships = document.querySelectorAll('.spaceship');
    for (let spaceship of spaceships) {
        let left = parseInt(spaceship.style.left) + xSpeed;
        spaceship.style.left = left + 'px';
    }
}

function spaceInvaders() {
    if (xDistance == 100) { // when the fleet hits the right boundary
        xSpeed = -1; // change direction
    }
    moveSpaceships();
    xDistance += xSpeed;
    requestAnimationFrame(spaceInvaders);
}

createGrid();
spaceInvaders();

```

Testing

- ☐ Testing the animation of the fleet. Your code has passed our test.

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:

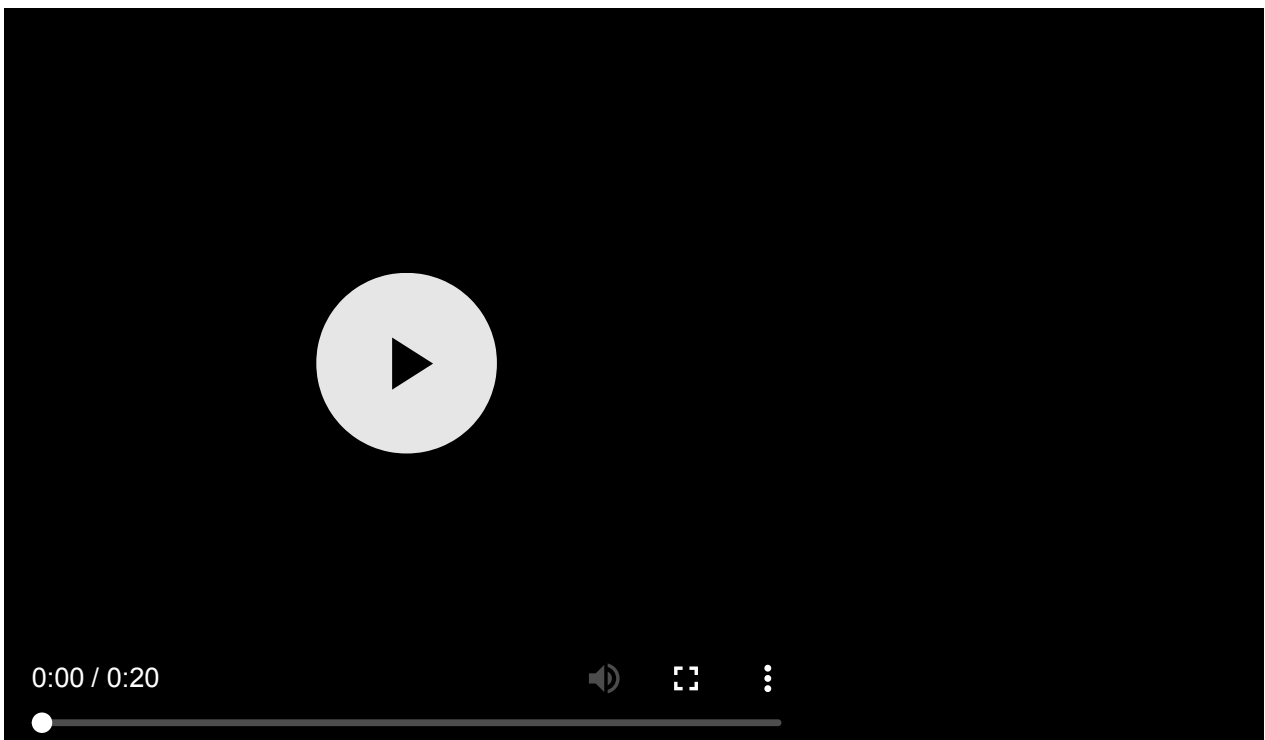
[Step 1: Make a new variable](#)

[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 something like this:



[The Zargan Fleet: Attack Formation \(https://groklearning-cdn.com/modules/VVVKujr4gqwwRYnrSMxZ5n/MissionBriefing3.mov\)](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.js

```

var gap = 10;
var spaceshipsPerRow = 10;
var startLeft = 30;
var startTop = 50;
var spaceshipWidth = 20;
var spaceshipHeight = 20;
var rowsOfSpaceships = 5;
var xSpeed = 1;
var xDistance = 1;

function createSpaceship(left, top) {
    let spaceship = document.createElement('img');
    spaceship.src = 'spaceship1.png';
    document.body.appendChild(spaceship);
    spaceship.className = 'spaceship';
    spaceship.style.top = top + 'px';
    spaceship.style.left = left + 'px';
}

function createRow(left, top) {
    for (let spaceshipCounter = 0; spaceshipCounter < spaceshipsPerRow; spaceshipCounter++) {
        createSpaceship(left, top);
        left = left + spaceshipWidth + gap;
    }
}

function createGrid() {
    let top = startTop;
    for (let row = 0; row < rowsOfSpaceships; row++) {
        createRow(startLeft, top);
        top = top + spaceshipHeight + gap;
    }
}

function moveSpaceships() {
    let spaceships = document.querySelectorAll('.spaceship');
    for (let spaceship of spaceships) {
        let left = parseInt(spaceship.style.left) + xSpeed;
        spaceship.style.left = left + 'px';
    }
}

function spaceInvaders() {
    if (xDistance == 100) { // when the fleet hits the right boundary
        xSpeed = -1; // change direction
    }
    if (xDistance == 0) { // when the fleet hits the left boundary
        xSpeed = 1; // change direction
    }

    moveSpaceships();
    xDistance += xSpeed;
    requestAnimationFrame(spaceInvaders);
}

createGrid();
spaceInvaders();

```

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?

```
function moveSpaceships() {
    let spaceships = document.querySelectorAll(".spaceship");
    for (let spaceship of spaceships) {
        let left = parseInt(spaceship.style.left) + 1;
        spaceship.style.left = left + 'px';
        let top = parseInt(spaceship.style.top) + 1;
        spaceship.style.top = top + 'px';
    }
    requestAnimationFrame(moveSpaceships);
}
```

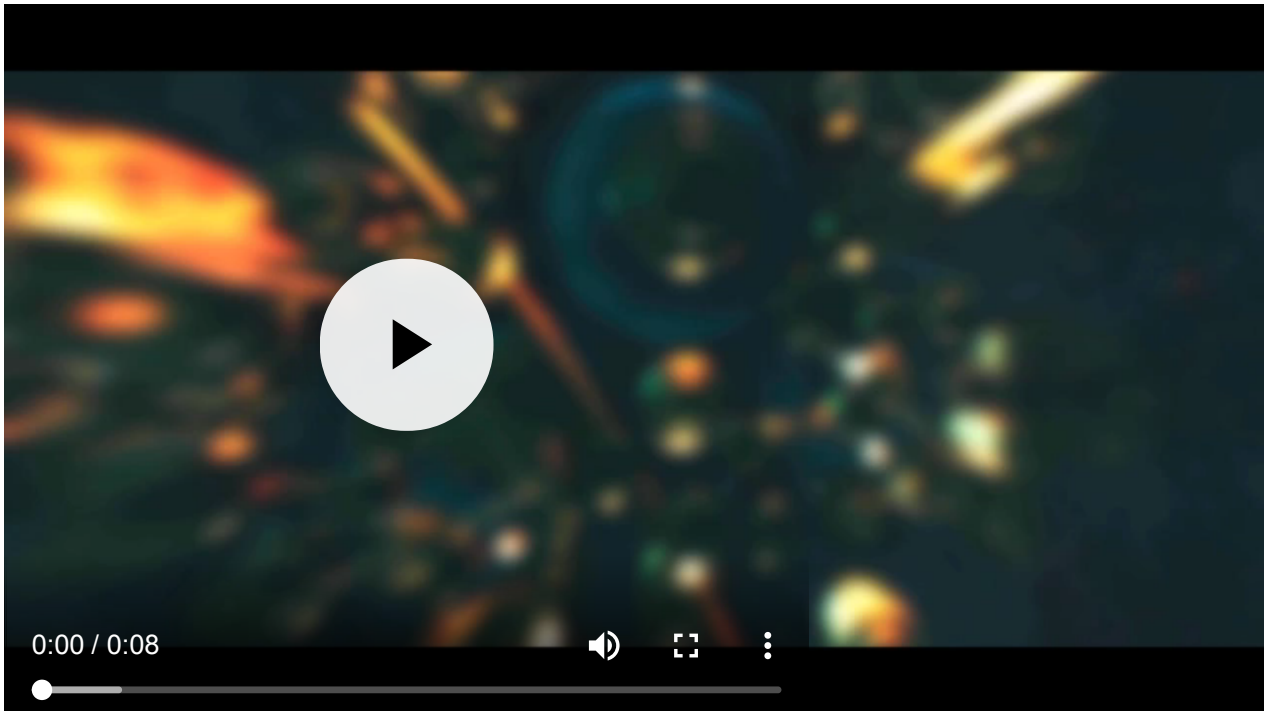
- ☐ Move spaceships in a diagonal line from top-left to bottom-right.
- ☐ Make spaceships jump around randomly
- ☐ Move spaceships in a horizontal line from right to left
- ☐ Nothing. Spaceships stay where they are.

Testing

☐ That's right!

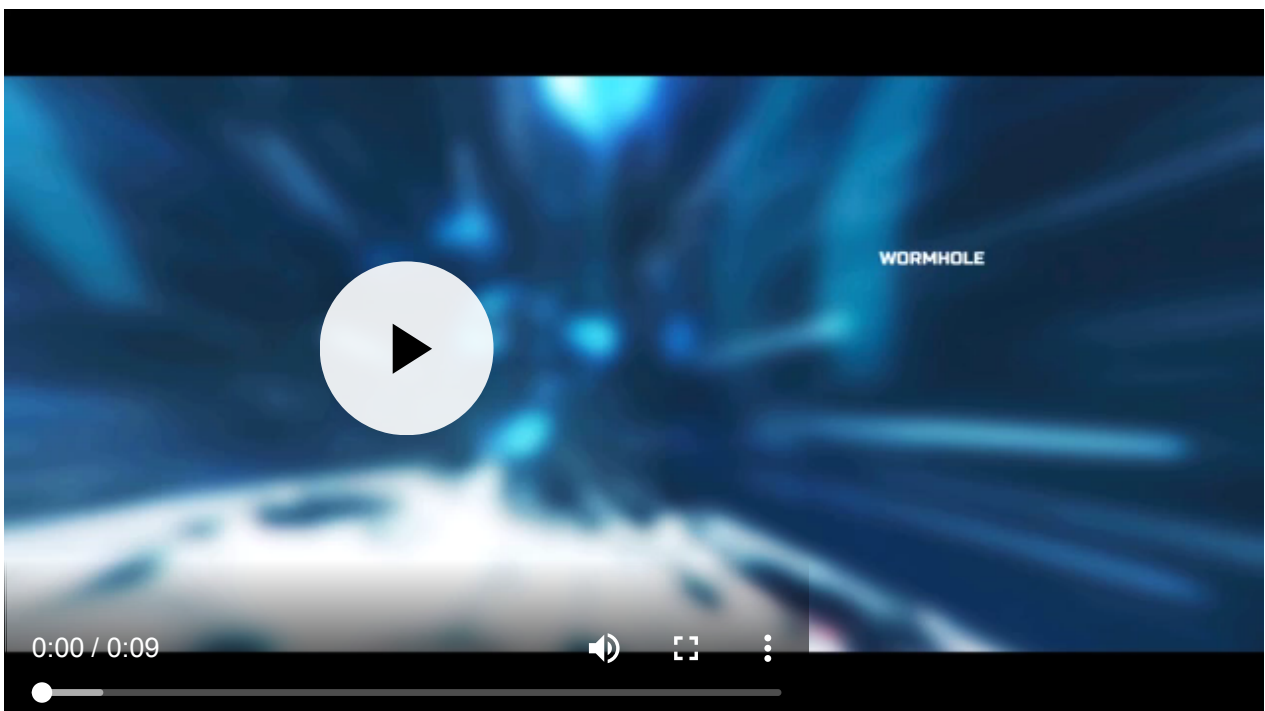
3.1.8. The cover is blown

Commodore, we have been spotted by Earth's early warning station deep inside the asteroid belt between Mars and Jupiter.



[Asteroid Base \(https://groklearning-cdn.com/modules/jzzWerNY6FHE4MXatyobH3/asteroidbase.mov\)](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.

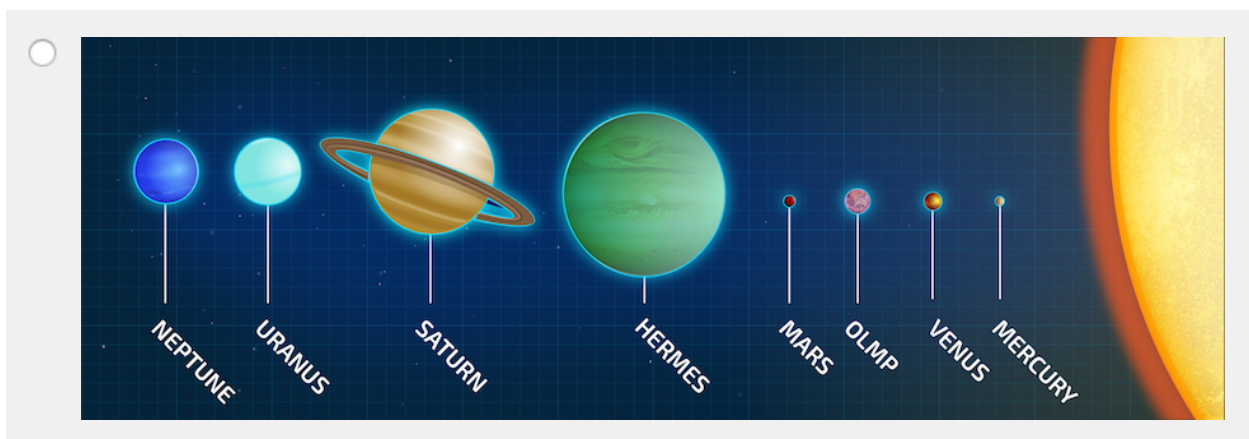
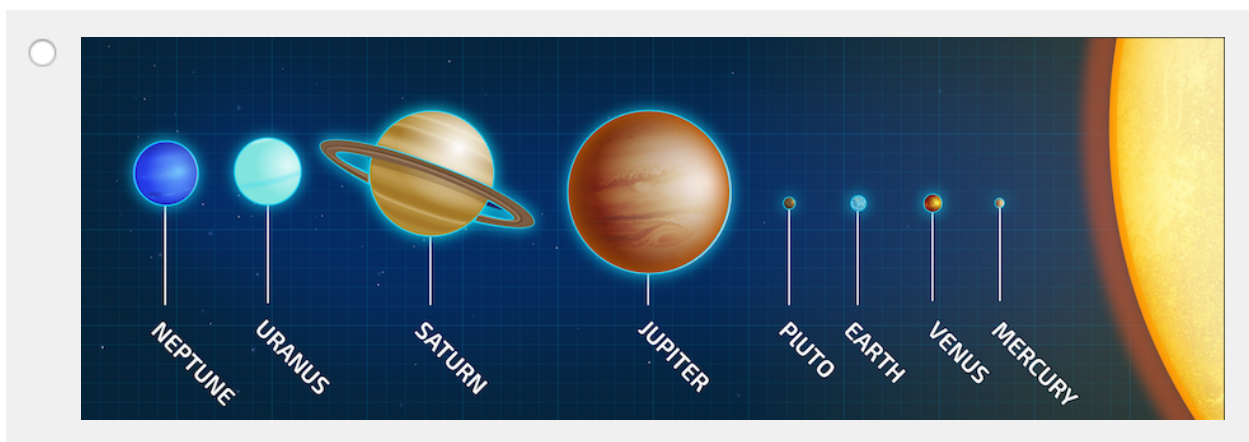
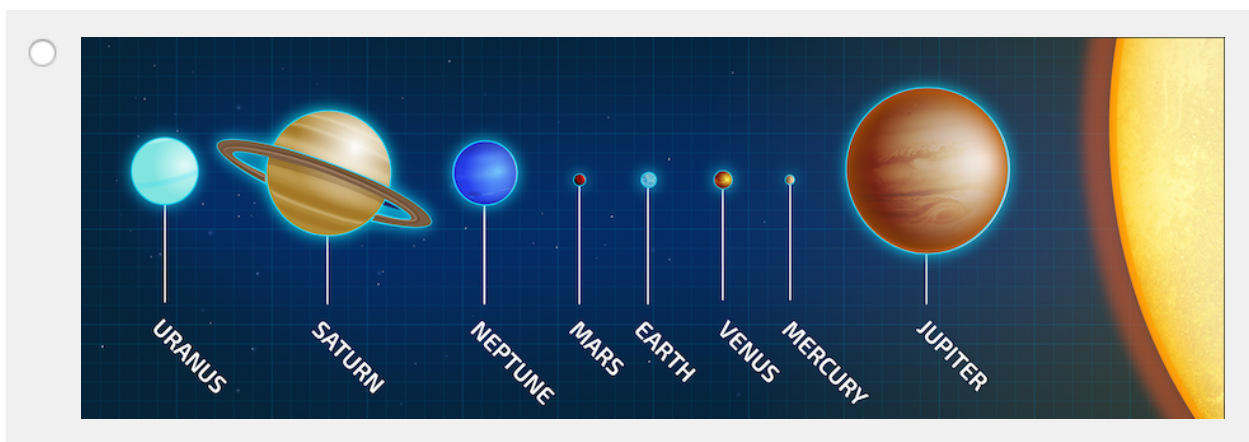
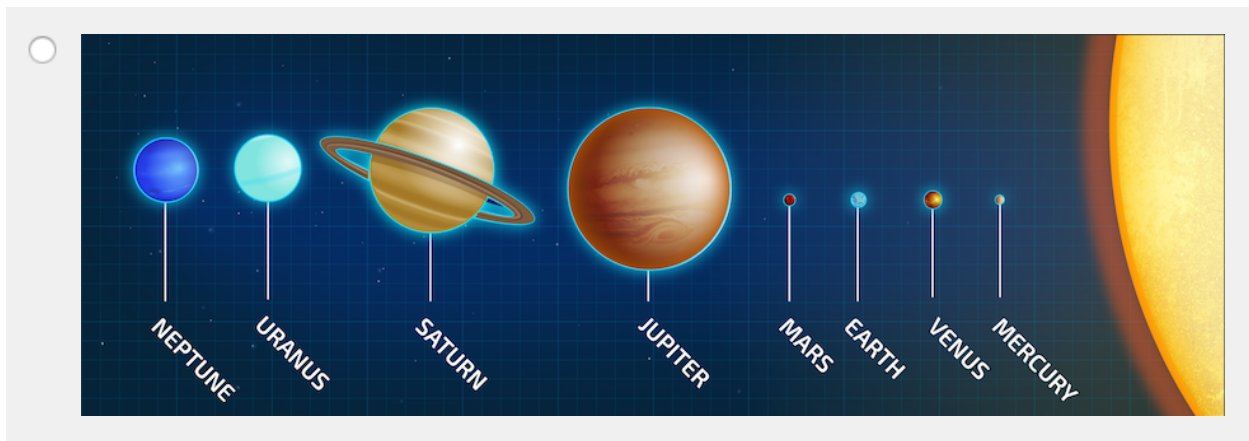


[Wormhole Drive \(https://groklearning-cdn.com/modules/z9MmVGrz32vqygcX3koQLa/wormhole.mov\)](https://groklearning-cdn.com/modules/z9MmVGrz32vqygcX3koQLa/wormhole.mov).

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.

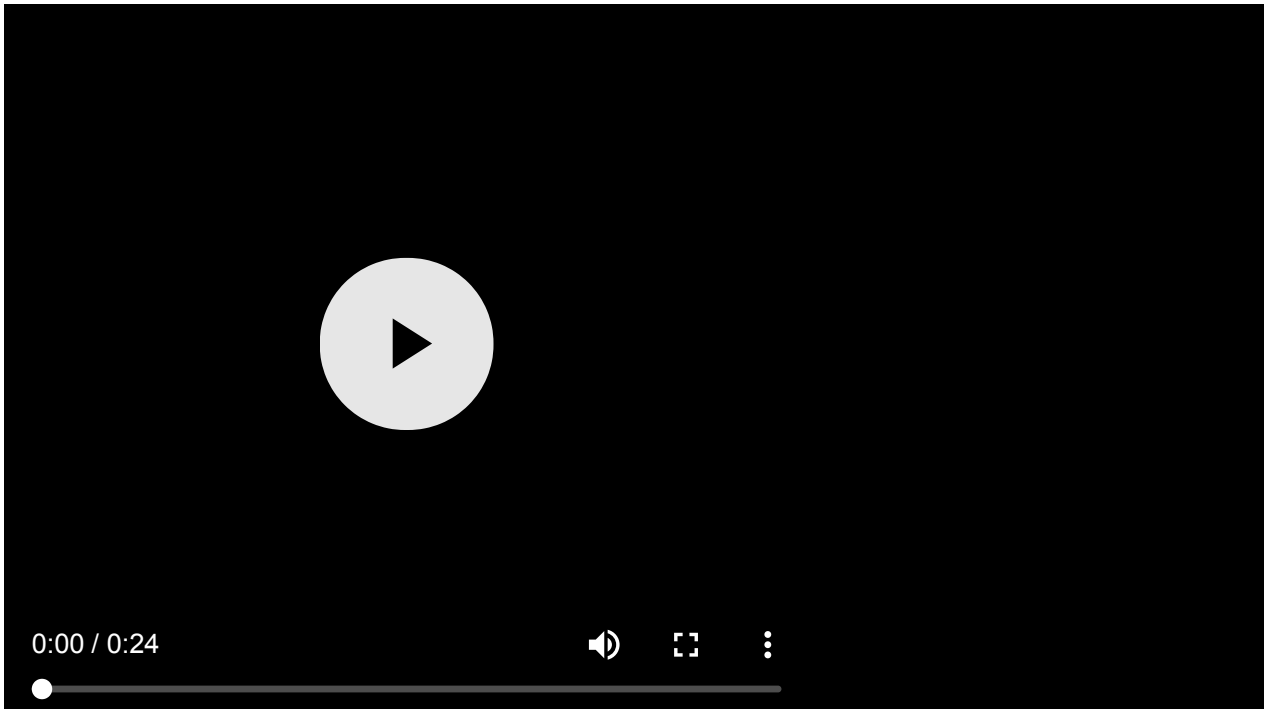
<https://aca.edu.au/challenges.html>

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.



[Earth \(https://groklearning-cdn.com/modules/SnkK98tT7QPLgef4gNAxQU/earth480p.mov\).](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

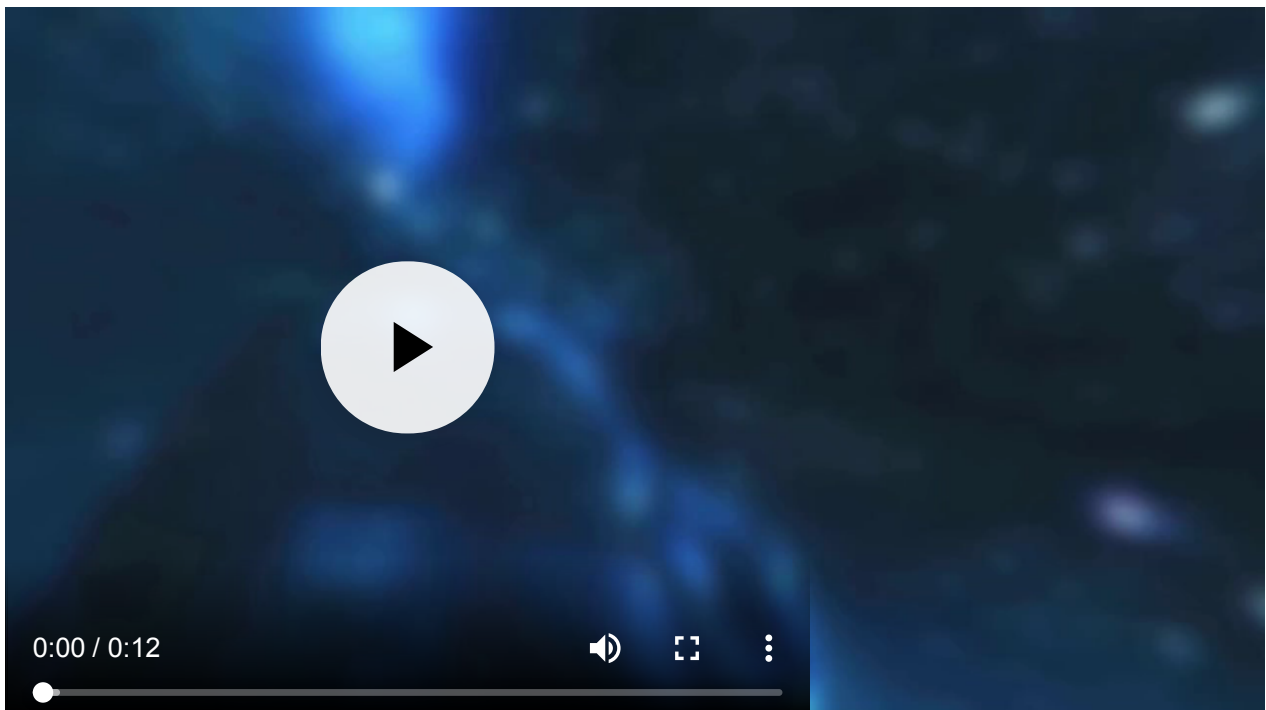
4.1.1. Mission briefing

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).



Red Alert ... Global Defence System Activated (<https://groklearning-cdn.com/modules/Aw8MiiXbZAKDbJPZyr26F5/gdsactivated480p.mov>).

4.1.2. Problem: Battle for Earth



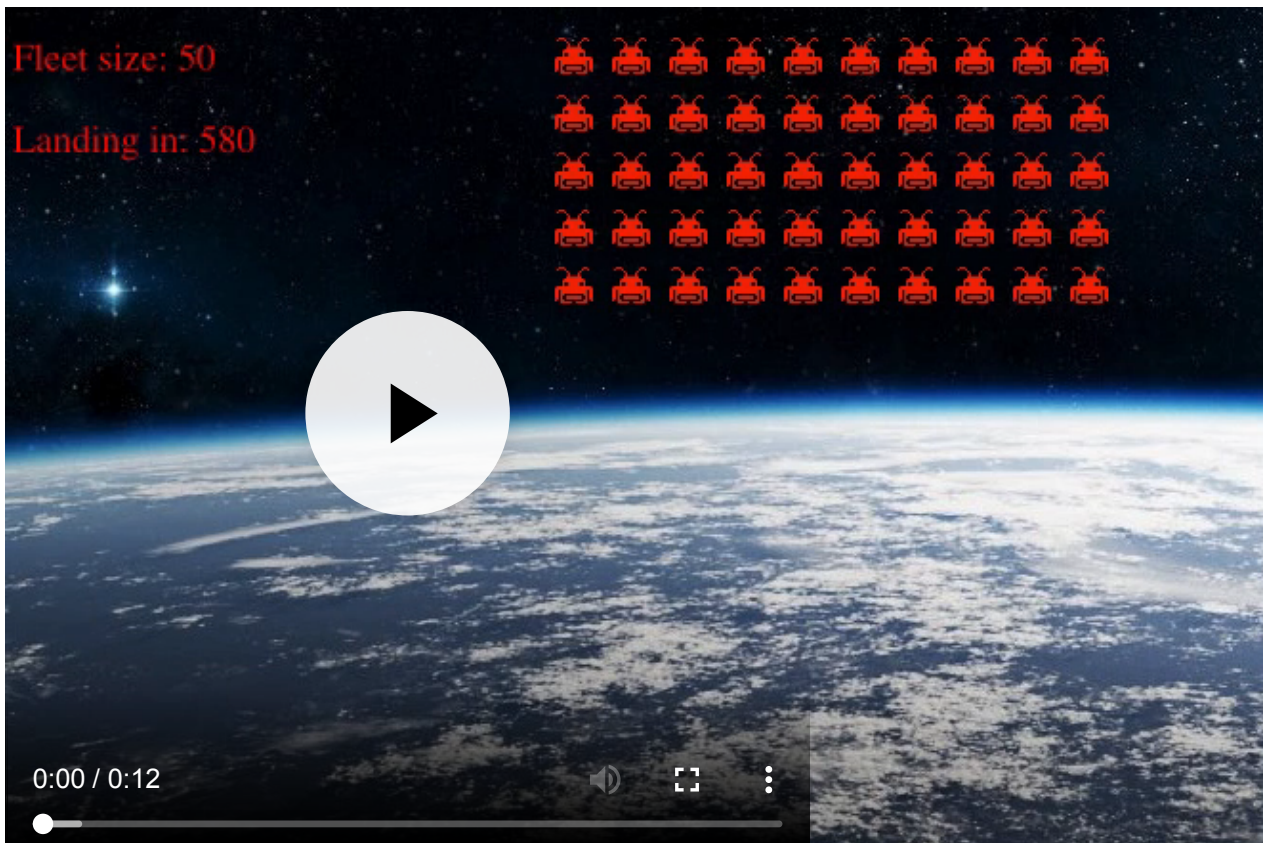
Rear Admiral, you are about to engage Earth's global defence system.

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** travelled 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 \(https://groklearning-cdn.com/modules/7788e3Ah3YVzstVhrfPrEL/battleForEarth.mov\)](https://groklearning-cdn.com/modules/7788e3Ah3YVzstVhrfPrEL/battleForEarth.mov).

You'll need

 program.js

```

var ySpeedMax = 10;
var gap = 10;
var spaceshipsPerRow = 10;
var startLeft = 150;
var startTop = 0;
var spaceshipWidth = 20;
var spaceshipHeight = 20;
var rowsOfSpaceships = 5;
var xSpeed = 1;
var ySpeed = 0;
var xDistance = 1;
var xDistanceMin = 300;
var rowNumber = 0;

function createSpaceship(left, top) {
    let spaceship = document.createElement('img');
    spaceship.src = 'spaceship' + (rowNumber % 7 + 1) + '.png'; // little trick to sh
    document.body.appendChild(spaceship);
    spaceship.className = 'spaceship';
    spaceship.style.top = top + 'px';
    spaceship.style.left = left + 'px';
}

function createRow(left, top) {
    for (let spaceshipCounter = 0; spaceshipCounter < spaceshipsPerRow; spaceshipCoun
        createSpaceship(left, top);
        left = left + spaceshipWidth + gap;
    }
    rowNumber++;
}

function createGrid() {
    let top = startTop;
    for (let row = 0; row < rowsOfSpaceships; row++) {
        createRow(startLeft, top);
        top = top + spaceshipHeight + gap;
    }
}

function moveSpaceships() {
    let spaceships = document.querySelectorAll('.spaceship');
    for (let spaceship of spaceships) {
        let left = parseInt(spaceship.style.left) + xSpeed;
        spaceship.style.left = left + 'px';
        let top = parseInt(spaceship.style.top) + ySpeed;
        spaceship.style.top = top + 'px';
    }
}

function spaceInvaders() {
    if (xDistance == 0 || xDistance == xDistanceMin) { // when the fleet hits the lef
        xSpeed *= -1; // change direction
        ySpeed = ySpeedMax; // move down
    }
    else {
        ySpeed = 0;
    }
    moveSpaceships();
    xDistance += xSpeed;
}

```

```
    requestAnimationFrame(spaceInvaders);  
}  
  
createGrid();  
spaceInvaders();
```

Testing

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

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.

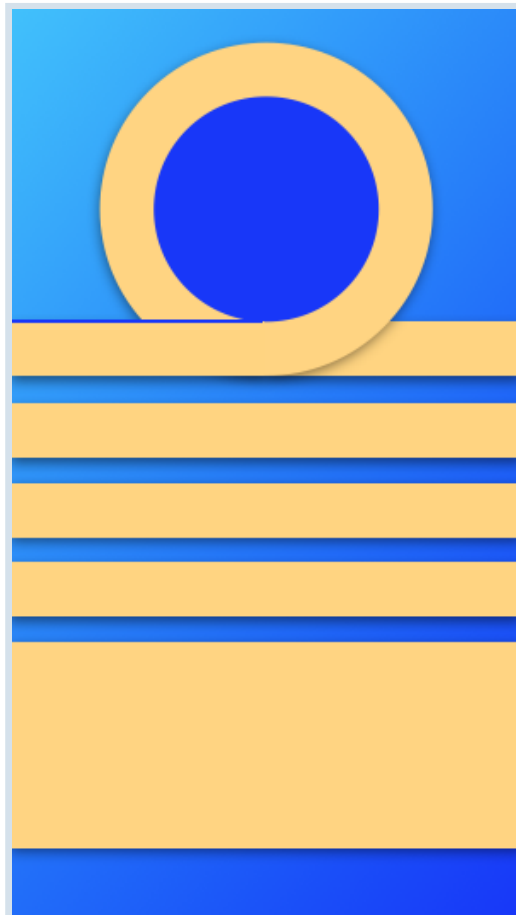
The Zargan High Command is very proud of your achievement and hereby promotes you to the highest rank in the Zargan Spacefleet:

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 \(https://groklearning-cdn.com/modules/9NZ5v38XB8NCaLKGtnTCbN/Fireworks480p.mov\).](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** calls the JavaScript files and sets the two fields to display the score and the distance to travel
- **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.js`

```

var ySpeedMax = 10;
var gap = 10;
var spaceshipsPerRow = 10;
var startLeft = 150;
var startTop = 0;
var spaceshipWidth = 20;
var spaceshipHeight = 20;
var rowsOfSpaceships = 5;
var xSpeed = 1;
var ySpeed = 0;
var xDistance = 1;
var xDistanceMin=300
var rowNumber = 0;

function createSpaceship(left, top) {
    let spaceship = document.createElement('img');
    spaceship.src = 'spaceship' + (rowNumber % 7 + 1) + '.png'; // little trick to sh
    document.body.appendChild(spaceship);
    spaceship.className = 'spaceship';
    spaceship.style.top = top + 'px';
    spaceship.style.left = left + 'px';
}

function createRow(left, top) {
    for (let spaceshipCounter = 0; spaceshipCounter < spaceshipsPerRow; spaceshipCoun
        createSpaceship(left, top);
        left = left + spaceshipWidth + gap;
    }
    rowNumber++;
}

function createGrid() {
    let top = startTop;
    for (let row = 0; row < rowsOfSpaceships; row++) {
        createRow(startLeft, top);
        top = top + spaceshipHeight + gap;
    }
}

function moveSpaceships() {
    let spaceships = document.querySelectorAll('.spaceship');
    for (let spaceship of spaceships) {
        let left = parseInt(spaceship.style.left) + xSpeed;
        spaceship.style.left = left + 'px';
        let top = parseInt(spaceship.style.top) + ySpeed;
        spaceship.style.top = top + 'px';
    }
}

function spaceInvaders() {
    if (xDistance == 0 || xDistance == xDistanceMin) { // when the fleet hits the lef
        xSpeed *= -1; // change direction
        ySpeed = ySpeedMax; // move down
    }
    else {
        ySpeed = 0;
    }
    moveSpaceships();
    xDistance += xSpeed;
}

```

```
    requestAnimationFrame(spaceInvaders);  
}  
  
createGrid();  
spaceInvaders();
```

 gds.js

```
//Global Defence System
```

```
var missileImage = 'https://groklearning-cdn.com/modules/4TQCjD2F82oHY2nhKbJ7L3/rocke
var collisionImage = 'https://groklearning-cdn.com/modules/wqPdjpNopABLR78oLLYCYU/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';
```

```

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) {

```

```

        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(distanceToTravel)
    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)) || setTimeout(
                tick, 1000 / 60);
        }
    };

    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)) || setTimeout(
                tick, 1000 / 60);
        }
    };
};

```

```

        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! Commencing water extraction. Victory !');
        }
    }

    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);
    }

```

 index.html

```
<html>
  <head>
    <title>Space Invaders</title>
    <link rel="stylesheet" href="stylesheet.css">
  </head>
  <body>
    <script type='text/javascript' src='program.js'></script>
    <script type='text/javascript' src='gds.js'></script>
  </body>
</html>
```

 stylesheet.css

[stylesheet.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 : 5%;
    width: 300px;
    z-index: 1000;
    border-radius: 10px;
    margin-left: -150px;
}

.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.