

You have two free stories left this month. [Upgrade for unlimited access](#)



# Build an Arcade Game With Vanilla JavaScript

Create a simple arcade game with DOM manipulation



Roberto Hernandez

[Follow](#)

Feb 25 · 4 min read ★



Photo by Carl Raw on Unsplash

Today's a special day for those who just are starting out in the

You have two free stories left this month. [Upgrade for unlimited access](#)



• • •

Despite this post being for beginners and newbies, it doesn't mean that more experienced developers won't find it useful too.

• • •

## What We Will Cover

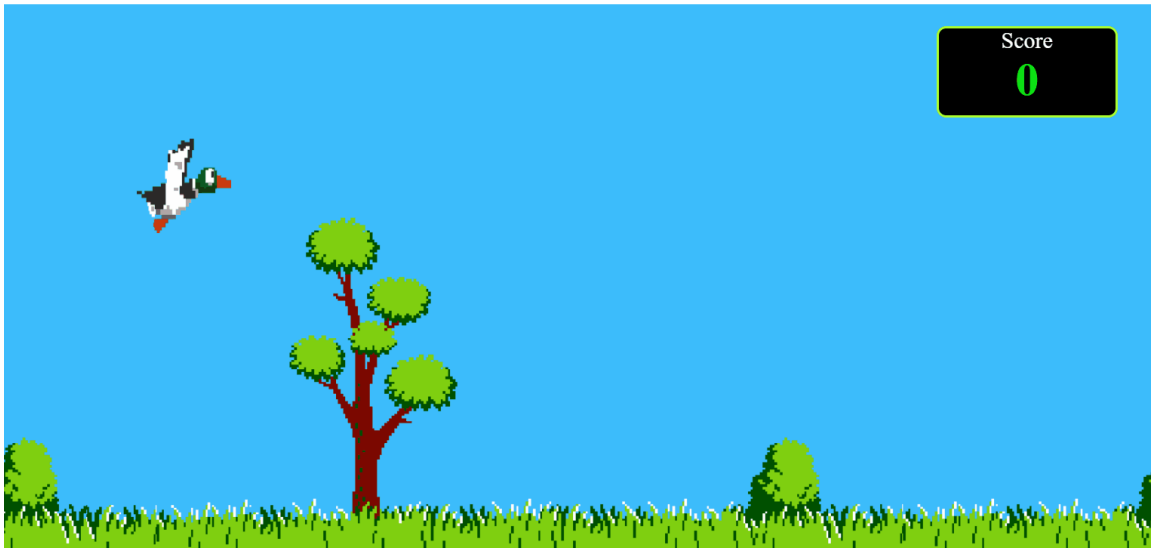
- The basics of CSS and JavaScript.
- The basic of Flexbox, the CSS3 web layout model.
- Dynamically manipulate the DOM using JavaScript.
- A walkthrough.

• • •

You have two free stories left this month. [Upgrade for unlimited access](#)

×

## The Challenge



Arcade game with vanilla JavaScript

The arcade game has the following requirements:

- It uses an image as a background and has a clickable duck image.
- When the user clicks on the duck two things happen: 1. The current score is increased by one and 2. The duck moves to a random position.
- It should be build using ES6 specification features.

You have two free stories left this month. [Upgrade for unlimited access](#)



• • •

You have two free stories left this month. [Upgrade for unlimited access](#)



# 1. Layout

Our layout (index.html) will have a div as a container and then both images the background and the duck. Finally, a `scoreContainer` element with the score text and the score (a counter).

```
<div class="container">
  
  
  <div class="scoreContainer">
    <div id="score-text">Score</div>
    <div id="score-counter">0</div>
  </div>
</div>
```

## Styles

```
/*Make any img element responsive*/
img {
  max-width: 100%;
}

/*Set a fixed size for width and height and in an
absolute position*/
#duck {
```

You have two free stories left this month. [Upgrade for unlimited access](#)



• • •

You have two free stories left this month. [Upgrade for unlimited access](#)



## 2. JavaScript

### 2.1 Create the event listener

Now, we are going to create an event listener on our duck image. When a user clicks on the duck image it will fire a function.

```
//Get the target element
const duck = document.querySelector("#duck");

//Add the click event listener
duck.addEventListener("click", () => {
  //Dont forget call the functions here
  increaseScore();
  moveDuck();
});
```

### 2.2 Create a function to increase the current score

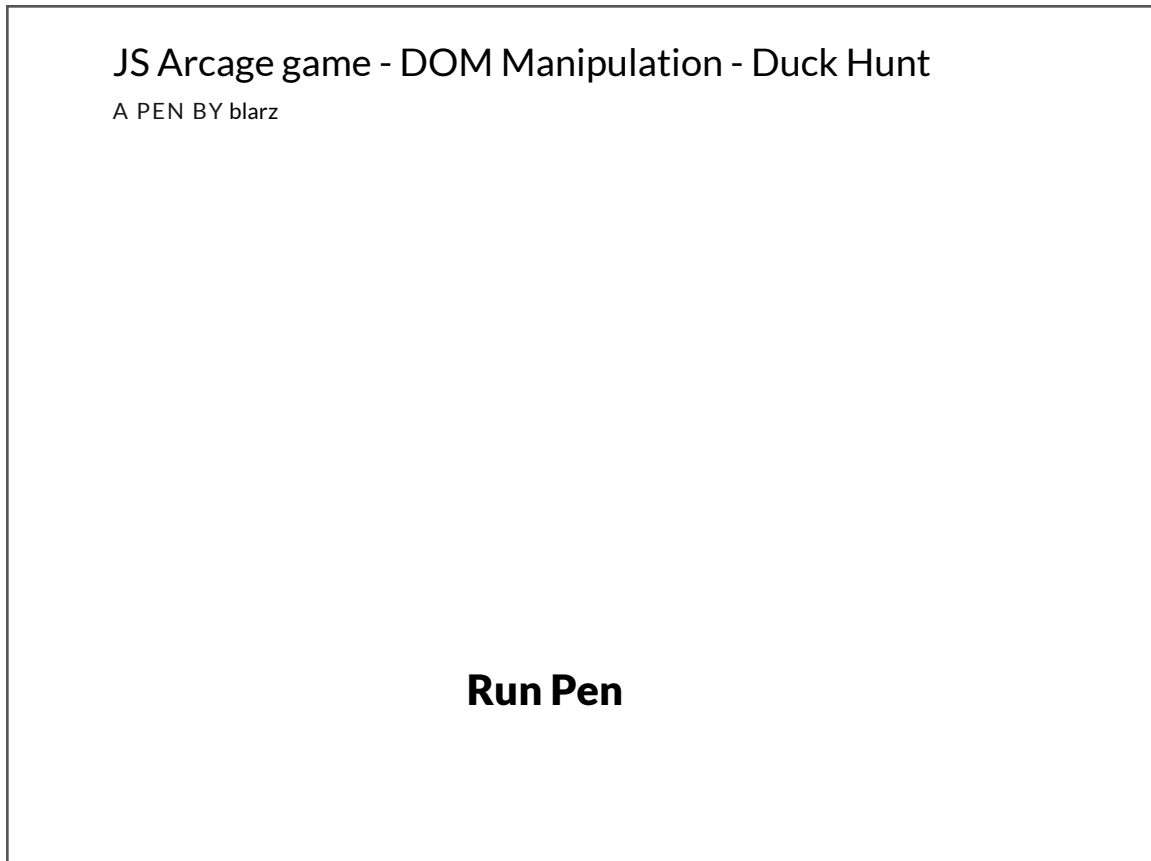
We just created the event listener. Now, we're going to create a function that will increase the counter, our score, by one.

```
//Increase score by 1
const increaseScore = () => {
  //Get the content of the target element. The current
  value for score
```



• • •

## The Completed Program



Arcade Game — DOM Manipulation

*Thanks for reading! I hope this post was helpful to you.*

Thanks to Zack Shapiro.

You have two free stories left this month. [Upgrade for unlimited access](#)



## Discover Medium

Welcome to a place where words matter. On Medium, smart voices and original ideas take center stage - with no ads in sight. [Watch](#)

## Make Medium yours

Follow all the topics you care about, and we'll deliver the best stories for you to your homepage and inbox. [Explore](#)

## Become a member

Get unlimited access to the best stories on Medium — and support writers while you're at it. Just \$5/month. [Upgrade](#)

[About](#)[Help](#)[Legal](#)

You have two free stories left this month. [Upgrade for unlimited access](#)

