

As every coding tutorial starts with a “Hello World” example, we’ll also follow the same. Our first tutorial on game development with *phaser.js* will be a “Hello World” example and a little extension over that to make you familiarize with basic functions of *phaser.js*. All our source code will be available in [github](#).

So now let’s start with our code. Our directory structure will be like below, where *js/phaser.min.js* file is the *phaser* APIs which you can download directly from our [github](#) page or from *phaser.io* website. *index.html* will be the source file that we’ll modify.

```
|
|_ assets
|   |_ images
|       |_ sky.png
|
|_ js
|   |_ phaser.min.js
|
|
|_ index.html
```

Now we’ll write our code inside *index.html* file. Just paste the following code inside this file.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Phaser Tutorial</title>
    <script type="text/javascript"
src="js/phaser.min.js"></script>
  </head>
  <body>

    <script type="text/javascript">

      var HEIGHT = 400;
      var WIDTH = 400;

      var game = new Phaser.Game(WIDTH, HEIGHT, Phaser.AUTO,
'hello-world-example', { preload: preload, create: create, update:
```

```

update });

        function preload() {
        }

        function create() {
        }

        function update() {
        }

    </script>

</body>
</html>

```

Line 13-14 : We are defining the area of our game space. In ideal case it should be full screen for your game. Later we'll see how to make it full screen.

Line 16 : We are defining the game object. This is the main content holder for all game related UI and other functions. While creating the game object, we've provided a few callback functions(preload, create and update). We'll learn more about these callback functions later.

Now you can run this program in your browser using any server. I'm using [mongosse server](#) as it is very lightweight. But you can use other server of your choice too. If you run your server at port 8080 and your source directory name is phaser-hello-world, use the URL <http://localhost:8080/phaser-hello-world/> to see the output in your browser.

The above piece of code only shows a black 400×400 rectangle on your browser. It doesn't do anything else. We'll add more pieces to it step by step. But before going into that let's talk about the callback functions that we provided in the *Phaser.Game* object.

- **preload**: This function is called by *phaser.js* before the game object is created. This is the function where you should load any resource required for your project. For example load image resource.
- **create**: called by *phaser.js* once the game object is initialized.
- **update**: this function is used to update the game state.

Now lets add a “Hello World” text in our game screen. Just add the following piece of code inside the *create* function.

```
var style = { font: "32px Arial", fill: "#ff0000", backgroundColor:
"#ffff00" };

game.add.text(0, 0, "Hello World", style);
```

Line #1 just creates a style for our font and *Line #3* adds the actual text at screen position (0,0) with the created style.

Next we’ll add an image background in our code. So load the required image inside the *preload* function as below.

```
function preload() {
    game.load.image('background', 'assets/images/sky.png');
}
```

Here we are loading the image and assigning it a key named ‘*background*’. Later we’ll use this key to draw the image. This is done by adding the following line of code at the beginning of the *create* function.

```
background = game.add.sprite(0, 0, 'background');
```

One important thing to note here is that, *phaser* will draw any UI object according to the sequence they are added. So if the above line of code is written at the end of *create* function, *phaser* will first draw the “Hello World” text and then draw the background image which will occlude the “Hello World” text. So when designing the game area, keep this in mind.

Now if the background image resolution is smaller than the resolution of the game area, you might want to scale up the image to cover the full game area. *phaser* has API for that too. So if our background image resolution is 100 x 100 and our game area is 400 x 400, we can use the following piece of code to scale up the image.

```
background = game.add.sprite(0, 0, 'background');
background.scale.setTo(4,4);
```

So that’s all for our “Hello World” example. In this tutorial we didn’t actually use the *update* method. Next we’ll add more cool topics where we’ll show how to use the *update*

method and step by step we'll develop a simple game. Don't forget to checkout the code from [github](#).