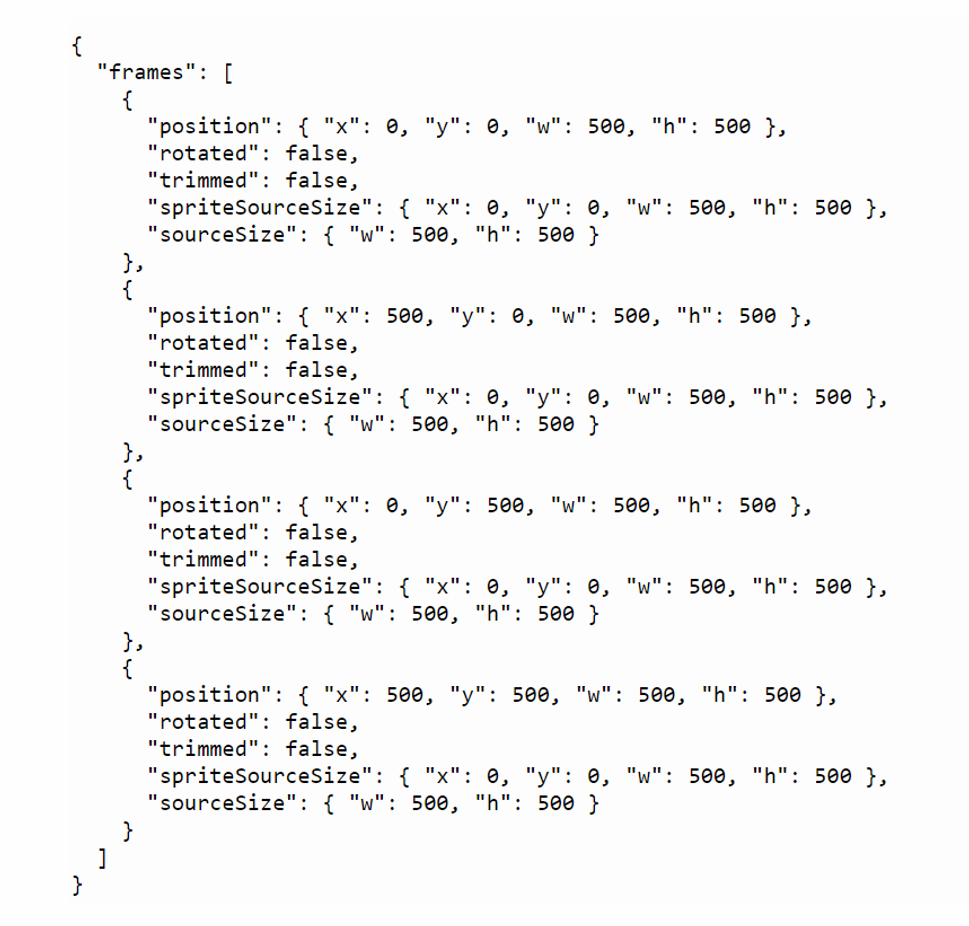
**Adding the animations** to the boats to give it a feel as if the boat is moving on the water

boat.json:



If you see here we have the main object and inside it, we have a **frames** key and its value is an **array** that contains multiple objects with different positions of the images. Looking at one example of the following we can see that an object has multiple keys such as position, rotated, trimmed, spriteSourceSize and sourceSize. The most important one is the position as it contains the position of the images.

We’ll first create a boatAnimation array. We’ll also declare two variables: boatSpritedata and boatSpriteSheet. boatSpritedata will contain data from JSON. boatSpritesheet will contain the images.

Now let’s get the data from the JSON and the image files. Now we’ll declare the boatFrames variable. This variable will contain the frame data in it.

create a boatFrames variable which contains the frames data in setup() of Sketch.js Using the for loop, we’ll loop on the length of the boatFrames. Inside this loop, we’ll get the position of each frame. From the SpriteSheet we’ll get the image with respective position and push this image to the boatAnimation array.

Write code to iterate through an array of boatFrames in setup() function in sketch.js file. variable pos is used to get the position of each frame from boatFrames. variable img is used to get the image from the boatSpritesheet which matches the position that we got in the pos variable and then pushing this image in the boatAnimation array.

**animate**()

We need the animation to move at a certain speed. So we’ll write an animate() function which will help us to set the speed of the animation. This speed will determine how fast every frame in our animation will move. The speed will keep on increasing as the game progresses and will be divided by the length of the animation so that every frame gets a certain amount of screen time. We will increment the speed by 0.05.

In the display() function we’ll create a variable index. We can use this variable to traverse through a set of animations. The index is calculated using the floor() method on this.speed & this.animation.length. Inside image() we’ll pass the animation.

**For broken boats:**

First we’ll create an empty array called brokenBoatAnimation and declare 2 variables brokenBoatSpritedata, brokenBoatSpriteSheet. And load the data to it. Creating an empty brokenBoatAnimation array and declaring the variables brokenBoatSpritedata and brokenBoatSpritesheet to store the frames and images of the broken boat

Using the for loop we’ll loop on the length of the boatFrames. Inside this loop we’ll get the position of each frame. From the SpriteSheet we’ll get the image with its respective position and push this image to the boatAnimation array

We had already written a function to remove the boats from the game. We want our boat to fade away slowly and we want to play our animation before the boat disappears. So for this we can use the function called setTimeout(). This function executes the code inside it after a certain interval of time.

Using the remove() function we are removing the boat of the given index from the world. As we want to remove the boat slowly. We have used the setTimeout function which calls the code inside it after the set intervals of time. We have set the time limit to 2 seconds. Inside the setTimeout() we have used the Matter.World.remove() function to remove the boat from the world and delete boats[index] to remove it from the array.

Similarly, the code for water splash when the cannonball hits the ground and see the splash animation when the cannonball hits the water