**imageMode(CENTER)**

We want our image to be drawn from the center.

In order to do that we are going to use the **imageMode(CENTER)** function.

This function tells our code to draw the images from the center which means when we pass the x and y position of the image, these points will be at the center of the image not at the top left corner.

This setting will be applied to all the images on the canvas including the background image.

Due to this the center of the background image will be at the origin of the canvas.

To prevent these issues we can use **push()** and **pop()** functions.

These functions are used when we want to apply new settings only for certain shapes or images.

**Angles in the p5.js library.**

Angles have 2 units-**radians** and **degrees**.

Just as we have different units for measuring length such as meter and feet.

Radians and degrees both are useful. But degrees is easy to work with because we have learned about the degrees in our math and drawing class as well.

But the problem is that in p5.js the unit of the angle is by **radians** by default.

So we have to change that to degrees.

For that purpose we have a function called **angleMode()**.

In this function we will write the name of the unit we want to use for angle.

In our case we will write **angleMode(DEGREES).**

This code we will write in the **setup()** function.

In p5.js angle increases in the clockwise direction

**translate()**

When we rotate any object in p5.js by default it rotates around the origin.

But in our case, we want to rotate the cannon at its origin point right?

To solve this problem we need to shift our origin at the position of the cannon.

For that we can use a function named **translate().**

This function translates (shift) the origin to the points given inside the brackets.

In our case, we want to shift the origin to the position of the cannon; so we need to pass this.x and this.y in the **translate()** function.

But this should be enclosed between the **push()** and **pop()** function otherwise we will shift the origin for all the bodies and we do not want that to happen.

There is one more important concept which we have to be aware of since we are shifting the origin to the position of the cannon then to create the cannon we will only give x and y as 0,0.

Because the origin itself came at the position of the cannon and we want to create the cannon there only.

If we write some other value then the cannon will be shifted to that amount in the x and y-direction.

**Set the velocity(speed) of the cannonball.**

**Matter.min.js** library has a function that will help us to set some velocity to the cannonball. The function is called **setVelocity()**.

This function takes the body to which we want to give velocity.

And the x & y velocity value as the parameters.

We want the cannonball to be shot from the ball at the angle that the cannon is pointing.

To do so **p5 library** has a predefined function **p5.Vector.fromAngle()**.

this function by default accepts the angle in **radians** but The angle we are providing is on degrees as our angle mode is in degrees, in order to pass the angle value to this function we need to **convert angle to radians.**

That is done by multiplying the angle value with (pi/180) which is (3.14/180).

This function creates and returns a 2D vector. Vector is just like an array. This vector contains x and y values.

A vector is a quantity that has a magnitude and the direction as well. Like we want to give velocity to our cannon ball. For that we need the amount of the velocity, that will be our magnitude and we also gave the direction in which our cannonball should move.

From this vector, we can get the x and y values and pass them to our cannonball.

We are already setting the angle to the cannonball by moving the cannon so we just need to get the velocity from the given cannon angle.

When we get the velocity from the function we’ll multiply the force using the multiply function as we want the cannonball to be launched with a great speed.

in the **setVelocity()** function the angle is in radians, but we need to convert that into degrees by multiplying with (180/pi) which is (180/3.14)