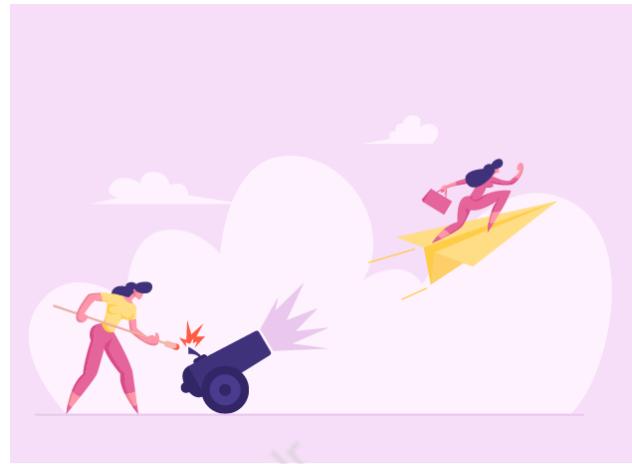


CANNON AND CANNONBALL



What is our GOAL for this MODULE?

In this class, we learned to create cannonballs for the Pirate Invasion game. We also created the moving cannon to adjust the shooting angle and shot the cannonballs.

What did we ACHIEVE in the class TODAY?

- Changed the angle of the cannon with arrow keys.
- Created the cannonball and shot it from the cannon.
- Created a function `shoot()` and set the velocity of the cannonball.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- Set different angles to the cannon.
- `setVelocity()`
- cannonball class
- `KeyIsDown()`

How did we DO the activities?

1. Create a **cannonball.js** file and add it to the index.html file.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Pirates Invasion</title>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.10.2/p5.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.10.2/addons/p5.sound.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/0.8.0/addons/p5.dom.min.js"></script>
    <script src="./lib/p5.gif.min.js"></script>
    <script src="./lib/matter.js"></script>

    <link rel="stylesheet" type="text/css" href="style.css" />

    <script src=".js/Cannon.js"></script>
    <script src=".js/CannonBall.js"></script>
  </head>
  <body>
    <script src="sketch.js"></script>
  </body>
</html>
```

2. Create the cannon class.

```
class CannonBall {
  constructor(x, y) {
    var options = {
      isStatic: true
    };
    this.r = 30;
    this.body = Bodies.circle(x, y, this.r, options);
    this.image = loadImage("./assets/cannonball.png");
    World.add(world, this.body);
  }
}
```

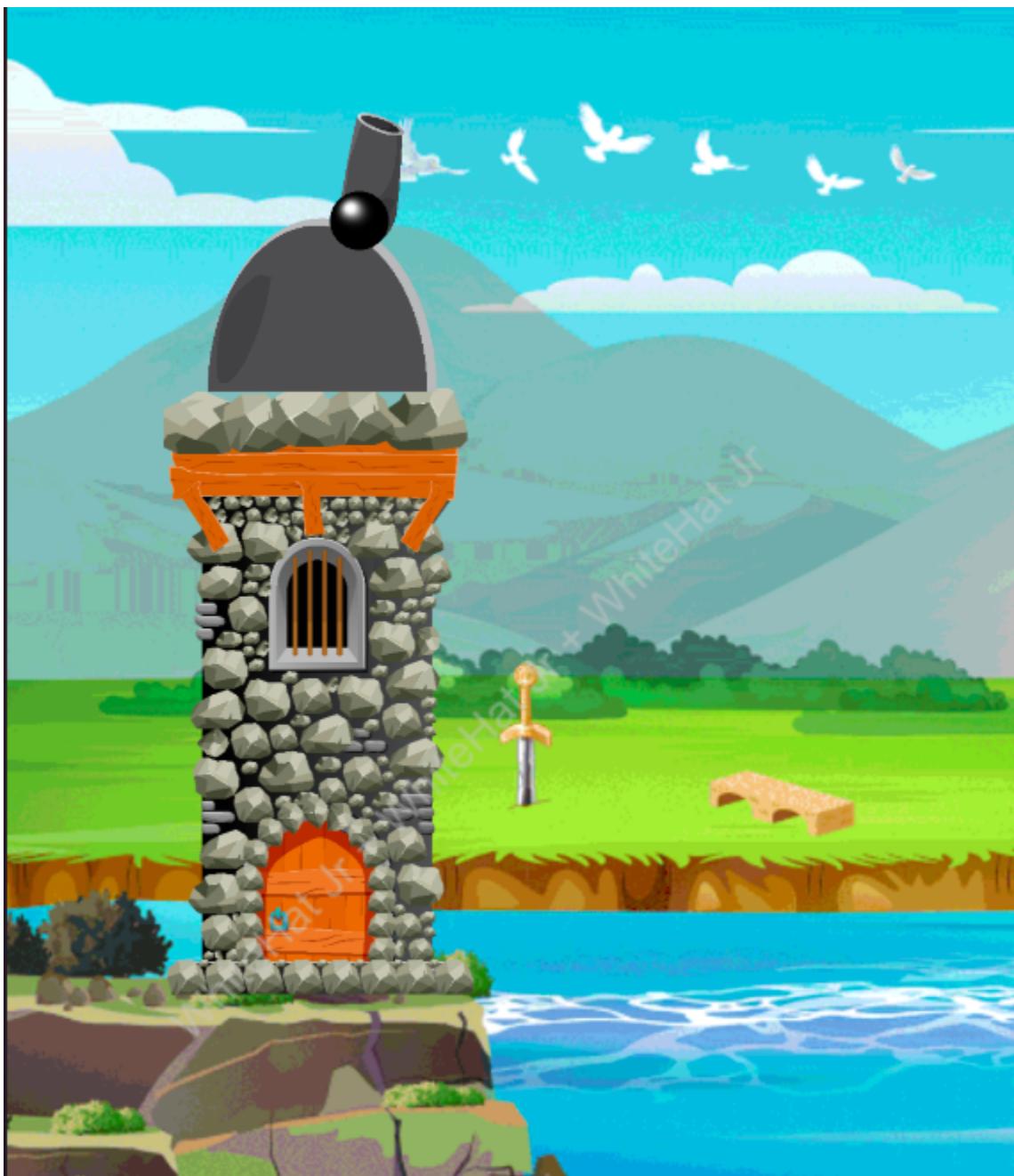
3. Use the **push()** and **pop()** function to apply new settings to the cannonball.
4. Create a new cannonball and display the cannonball in the **sketch.js** file.

```
display()
{
    var pos = this.body.position;
    push();
    imageMode(CENTER);
    image(this.image, pos.x, pos.y, this.r, this.r);
    pop();
}
```

```
cannon = new Cannon(180, 110, 130, 100, angle);
cannonBall = new CannonBall(cannon.x, cannon.y);
```

```
cannon.display();
cannonBall.display();
```

Output:



5. Change the angle of the cannon when the left and right arrow keys are pressed by adding **this.angle** property in the **constructor()** for the cannon.

```
class Cannon {  
    constructor(x, y, width, height, angle) {  
        this.x = x;  
        this.y = y;  
        this.width = width;  
        this.height = height;  
        this.angle = angle; this.angle = angle;  
        this.cannon_image = loadImage("assets/canon.png");  
        this.cannon_base = loadImage("assets/cannonBase.png");  
    }  
}
```

6. The angle in p5 is measured in radian by default. To change the angle in degrees, Use the **angleMode()** function.

```
function setup() {
    canvas = createCanvas(1200, 600);
    engine = Engine.create();
    world = engine.world;

    angleMode(DEGREES);
    angle = 15;

    ground = Bodies.rectangle(0, height - 1, width * 2, 1, { isStatic: true });
    World.add(world, ground);

    tower = Bodies.rectangle(160, 350, 160, 310, { isStatic: true });
    World.add(world, tower);

    cannon = new Cannon(180, 110, 130, 100, angle);
    cannonBall = new CannonBall(cannon.x, cannon.y);
}
```

7. Move the angle of the cannon up and down by using the **UP_ARROW** and **DOWN_ARROW** keys.

```
display() {
    console.log(this.angle)
    if (keyIsDown(RIGHT_ARROW)) {
        this.angle += 1;
    }

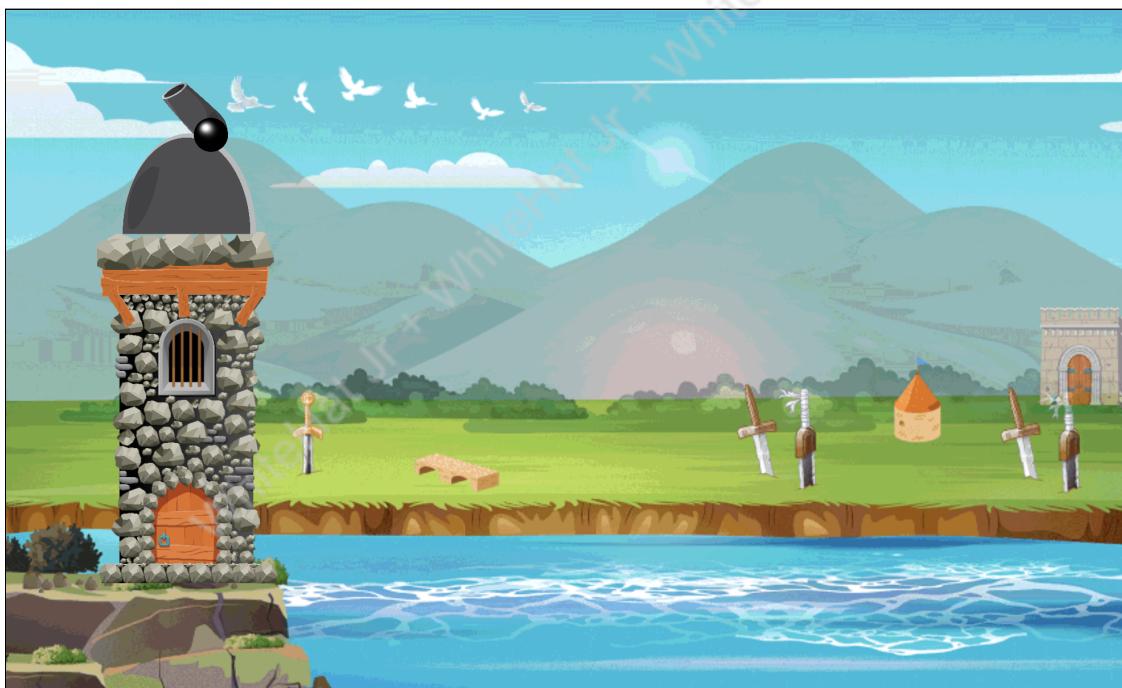
    if (keyIsDown(LEFT_ARROW)) {
        this.angle -= 1;
    }

    push();
    rotate(this.angle);
    imageMode(CENTER);
    image(this.cannon_image, this.x, this.y, this.width, this.height);
    pop();
    image(this.cannon_base, 70, 20, 200, 200);
    noFill();
}
}
```

8. Move the cannon with a given angle and not from the origin by using the **translate()** function.

```
push();
translate(this.x, this.y);
rotate(this.angle);
imageMode(CENTER);
image(this.cannon_image,0,0, this.width, this.height);
pop();
image(this.cannon_base, 70, 20, 200, 200);
noFill();
}
```

Output:



9. Restrict the cannon from moving in wide-angle by adding a condition to the earlier condition where we are changing the angle of the cannon on keypress.

```
display() {  
    console.log(this.angle)  
    if (keyIsDown(RIGHT_ARROW) && this.angle<70 ) {  
        this.angle += 1;  
    }  
  
    if (keyIsDown(LEFT_ARROW) && this.angle>-30 ) {  
        this.angle -= 1;  
    }  
}
```

Output:



10. Create the cannonball and shoot it from the cannon.
 - Use the **Bodies.circle()** to create the circular bodies.

11. Create a new cannonball object in **Sketch.js**.

```
cannonBall = new CannonBall(cannon.x, cannon.y);
```

12. Create a function **shoot()** in **cannonBall.js**.

- set the velocity to the cannonball using **SetVelocity()**.

```
shoot() {  
    Matter.Body.setStatic(this.body, false);  
    Matter.Body.setVelocity(this.body, { x: 30, y: -20 });  
}
```

13. Call the **cannonBall.shoot()** inside sketch.js after the function **draw()** using the **keyPressed()** function.

```
function keyReleased() {  
    if (keyCode === DOWN_ARROW) {  
        cannonBall.shoot();  
    }  
}
```

14. Use **p5.Vector.fromAngle()** function to align angle of cannonBalls to that of with cannon

- Modify the code in **cannonBall.js** for **shoot()**.
- Store the angle of canon in a variable **velocity**.
- Multiply that value with 0.5 using **mult()**.
- Change the **isStatic** property of cannonBall to false, so that it can fall.
- Set velocity using **setVelocity()** and **velocity** variables.

```
shoot() {  
    var newAngle = cannon.angle - 28;  
    newAngle = newAngle*(3.14/180)  
    var velocity = p5.Vector.fromAngle(newAngle);  
    velocity.mult(0.5);  
    Matter.Body.setStatic(this.body, false);  
    Matter.Body.setVelocity(this.body, {  
        x: velocity.x * (180 / 3.14),  
        y: velocity.y * (180 / 3.14) });  
}
```

OUTPUT:



What's next?

In the next class, we will learn how to create multiple balls and shoot them from the cannon.