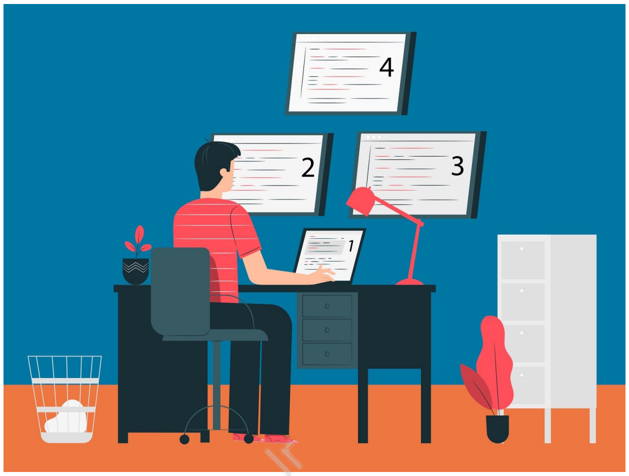
# INHERITANCE



## What is our GOAL for this MODULE?

We used our knowledge about inheritance to create objects with properties of parent class.

## What did we ACHIEVE in the class TODAY?

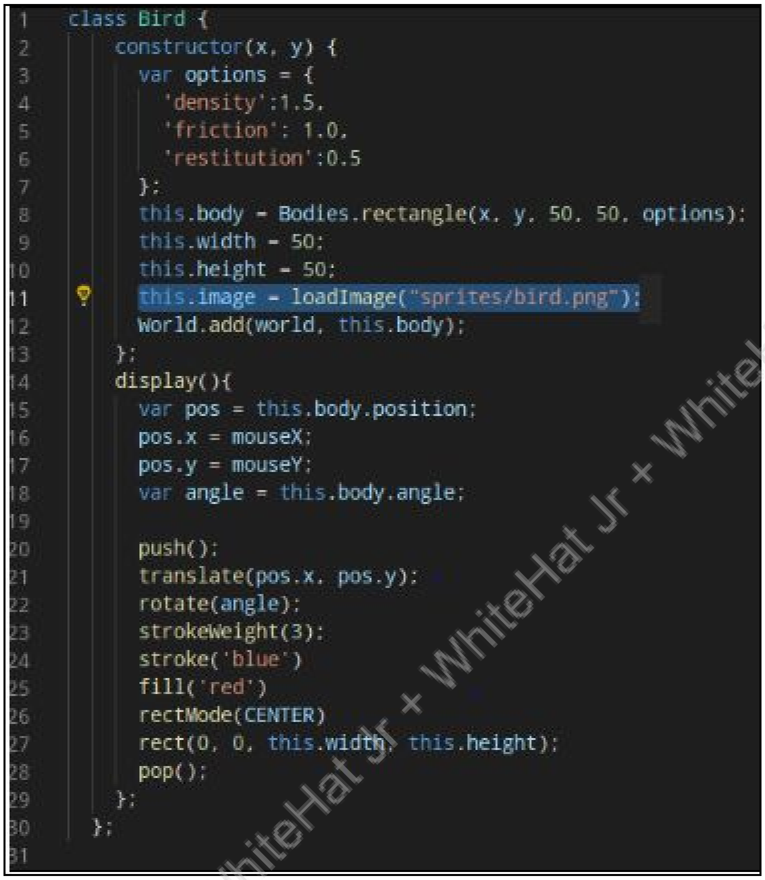
- Added an image property to one of the classes and used it to make one of the game objects animated.
- Introduced the concept of inheritance and how a subclass can inherit the properties and functions of a parent class.
- Wrote a subclass which extends the properties and functions of a parent class.
- Revised the following as well:
    - A class is a blueprint of an object.
    - A class contains defined properties (like width, height) and functions (like **display()**).
    - A class is used to create one or more objects having the same properties and functions as defined in the class.

## Which CONCEPTS/ CODING BLOCKS did we cover today?

- The concept of inheritance.
- extends keyword.
- Overriding method.

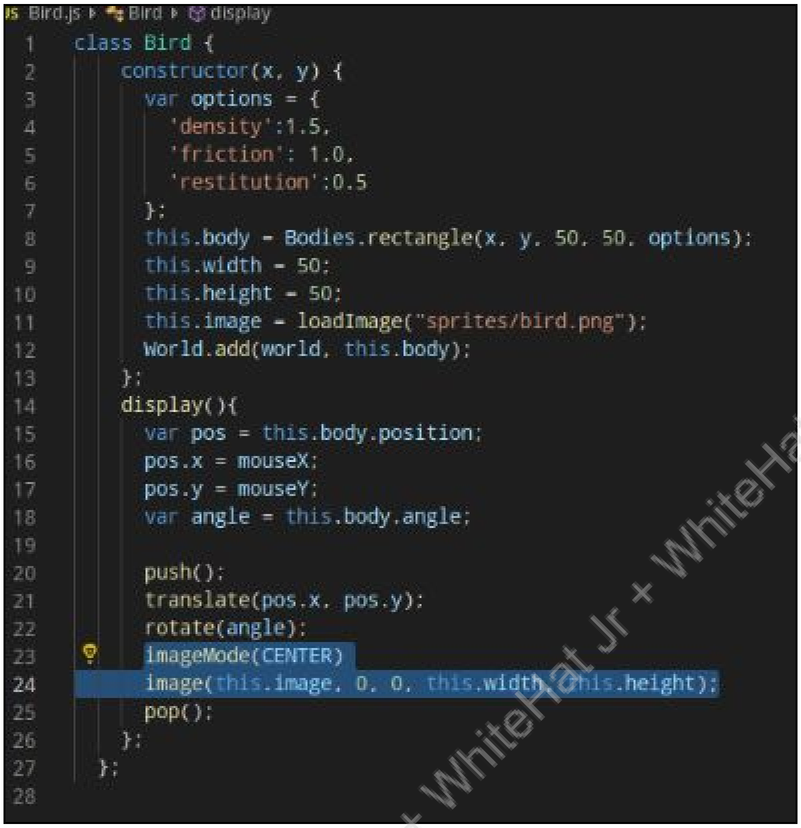
## How did we DO the activities?

1. Add an additional property called image and load the bird image to the existing Bird class.

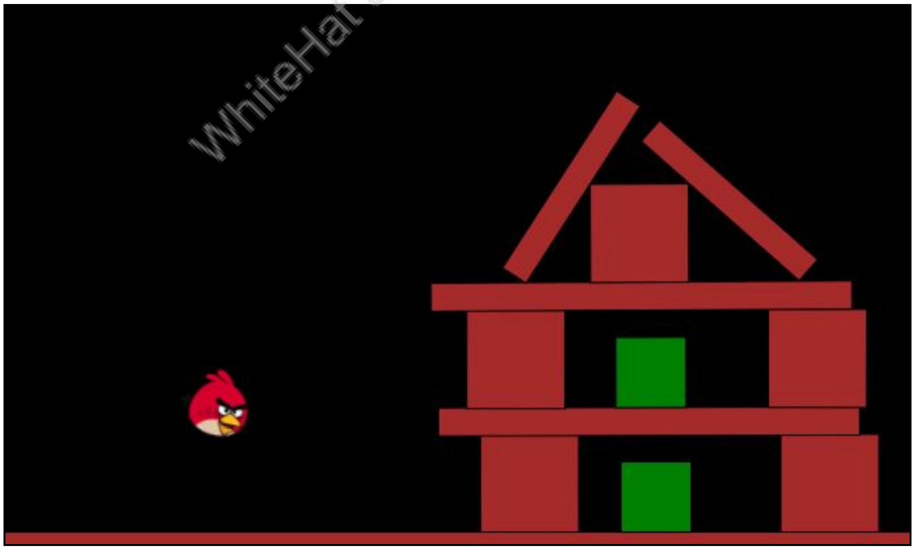


```
class Bird {
    constructor(x, y) {
      var options = {
        'density':1.5,
        'friction': 1.0,
        'restitution':0.5
      };
      this.body = Bodies.rectangle(x, y, 50, 50, options);
      this.width = 50;
      this.height = 50;
      this.image = loadImage("sprites/bird.png");
      World.add(world, this.body);
    };
    display(){
      var pos = this.body.position;
      pos.x = mouseX;
      pos.y = mouseY;
      var angle = this.body.angle;

      push();
      translate(pos.x, pos.y);
      rotate(angle);
      strokeWeight(3);
      stroke('blue')
      fill('red')
      rectMode(CENTER)
      rect(0, 0, this.width, this.height);
      pop();
    };
  };
```

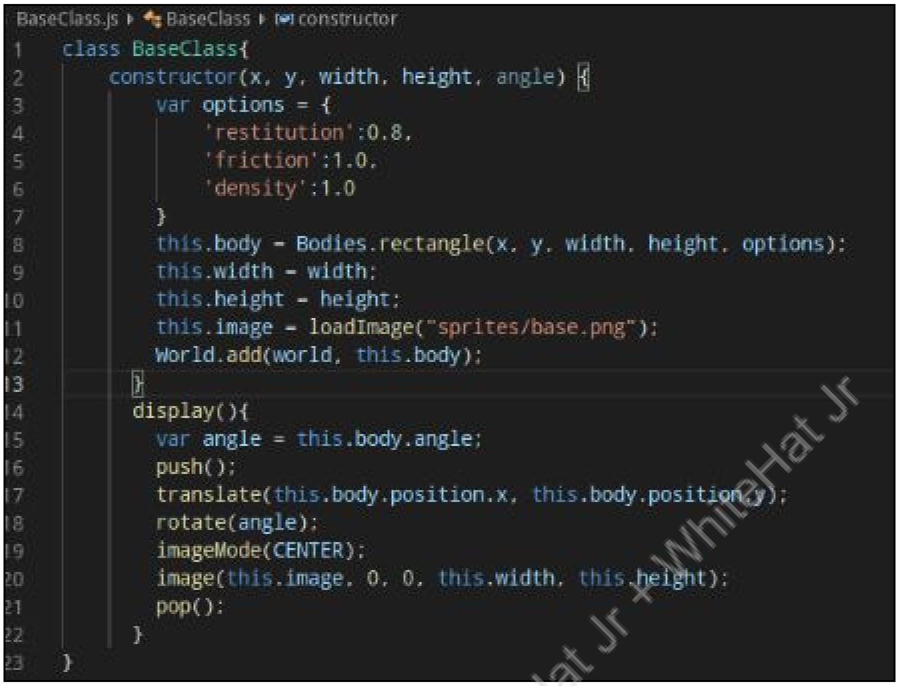
2. Change  image instead of a rectangle. Use the **image()** instruction instead of the **rect()** instruction.



```
class Bird {
    constructor(x, y) {
      var options = {
        'density':1.5,
        'friction': 1.0,
        'restitution':0.5
      };
      this.body = Bodies.rectangle(x, y, 50, 50, options);
      this.width = 50;
      this.height = 50;
      this.image = loadImage("sprites/bird.png");
      World.add(world, this.body);
    };
    display(){
      var pos = this.body.position;
      pos.x = mouseX;
      pos.y = mouseY;
      var angle = this.body.angle;

      push();
      translate(pos.x, pos.y);
      rotate(angle);
      imageMode(CENTER)
      image(this.image, 0, 0, this.width, this.height);
      pop();
    };
  };
```

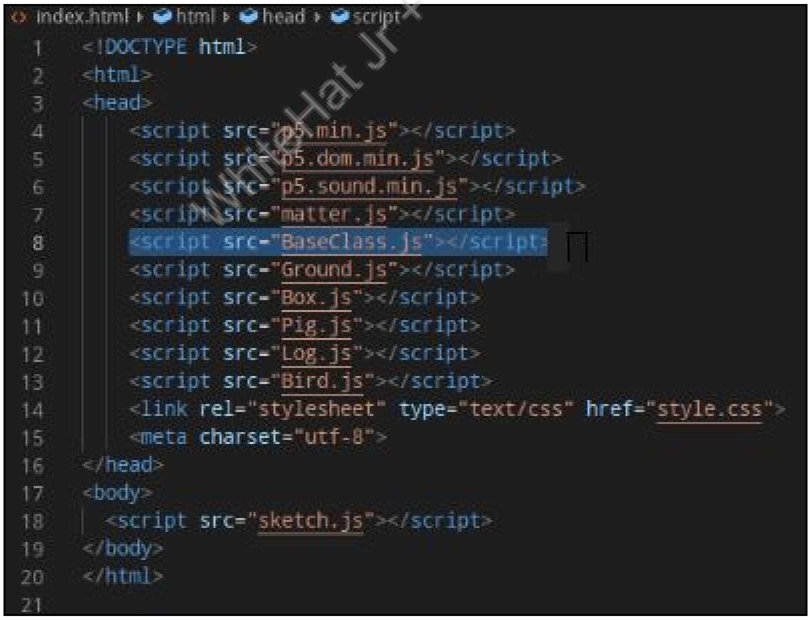


3. Create a **BaseClass** with all the properties and functions which we had in the Bird class.



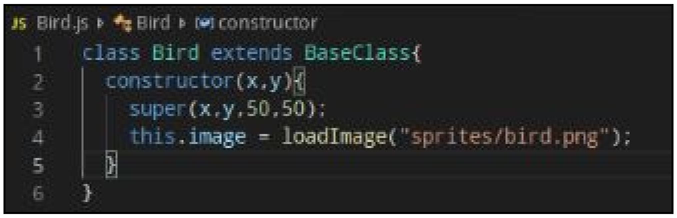
```
class BaseClass{
    constructor(x, y, width, height, angle) {
        var options = {
            'restitution':0.8,
            'friction':1.0,
            'density':1.0
        }
        this.body = Bodies.rectangle(x, y, width, height, options);
        this.width = width;
        this.height = height;
        this.image = loadImage("sprites/base.png");
        World.add(world, this.body);
      }
      display(){
        var angle = this.body.angle;
        push();
        translate(this.body.position.x, this.body.position.y);
        rotate(angle);
        imageMode(CENTER);
        image(this.image, 0, 0, this.width, this.height);
        pop();
      }
}
```

4. Included the src of the **BaseClass** in the **index.html** file.



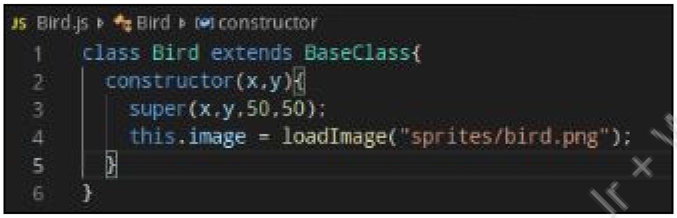
```
<!DOCTYPE html>
<html>
<head>
    <script src="p5.min.js"></script>
    <script src="p5.dom.min.js"></script>
    <script src="p5.sound.min.js"></script>
    <script src="matter.js"></script>
    <script src="BaseClass.js"></script>
    <script src="Ground.js"></script>
    <script src="Box.js"></script>
    <script src="Pig.js"></script>
    <script src="Log.js"></script>
    <script src="Bird.js"></script>
    <link rel="stylesheet" type="text/css" href="style.css">
    <meta charset="utf-8">
</head>
<body>
  <script src="sketch.js"></script>
</body>
</html>
```

5. Create a child Bird class which inherits all the properties and functions of our BaseClass.



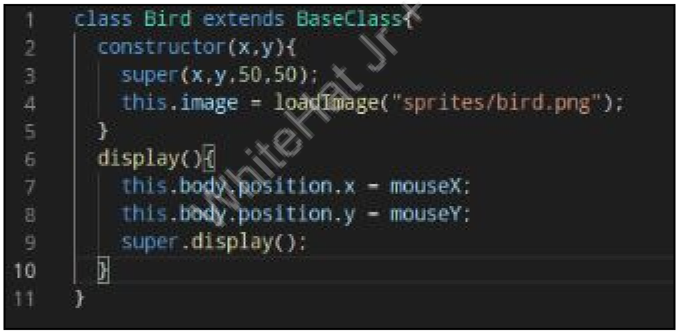
```
class Bird extends BaseClass{
  constructor(x,y){
    super(x,y,50,50);
    this.image = loadImage("sprites/bird.png");
  }
}
```

6. Add the bird image to the Bird class constructor as well. You could do it inside the constructor and overwrite any of the properties of the parent class inside the child class and change it.

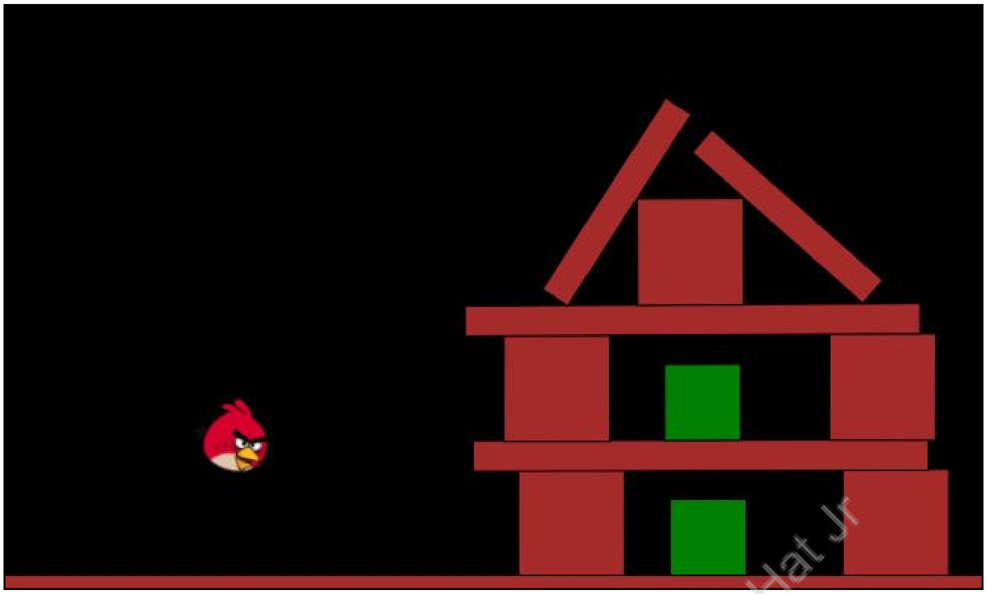


```
class Bird extends BaseClass{
  constructor(x,y){
    super(x,y,50,50);
    this.image = loadImage("sprites/bird.png");
  }
}
```

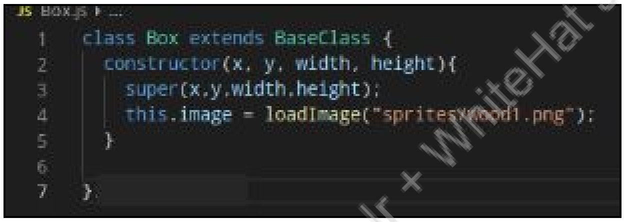
7. Override the display function of the base class by writing code for it, we used **super.display()** to refer to the parent class display function.



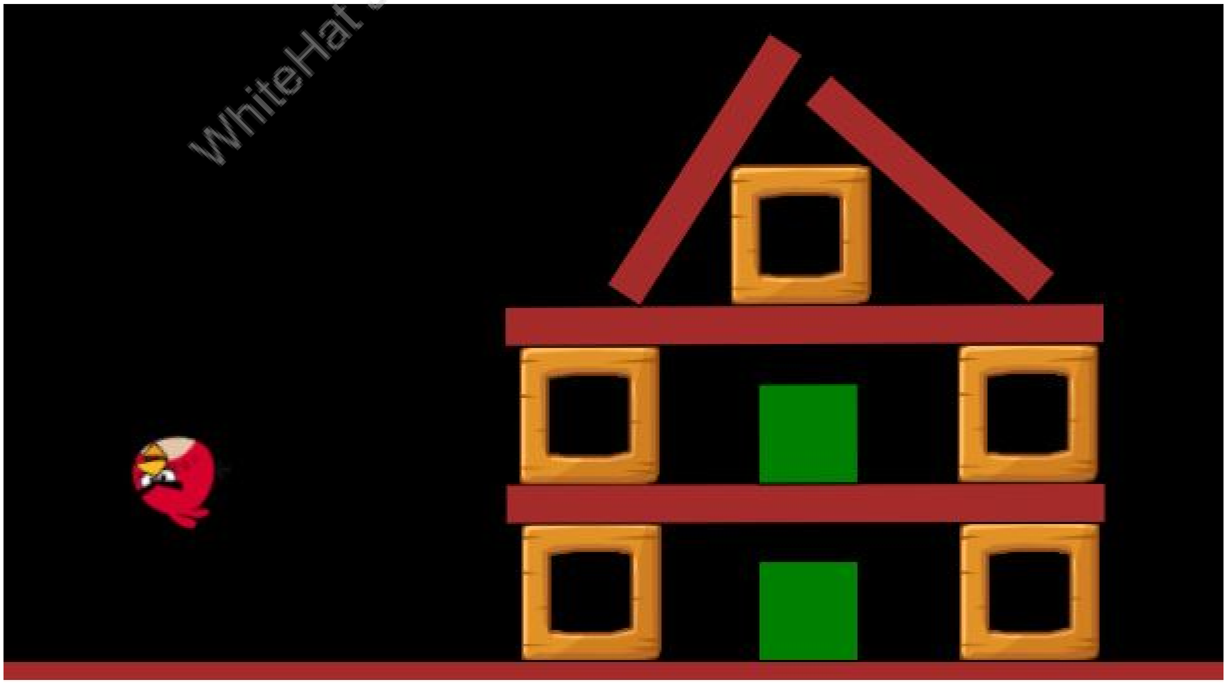
```
class Bird extends BaseClass{
  constructor(x,y){
    super(x,y,50,50);
    this.image = loadImage("sprites/bird.png");
  }
  display(){
    this.body.position.x = mouseX;
    this.body.position.y = mouseY;
    super.display();
  }
}
```



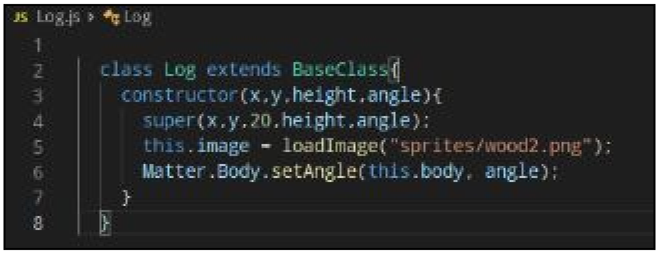
8. Inherit the Box class similar to Bird and add images to all the other objects in the game by modifying their class blueprint.



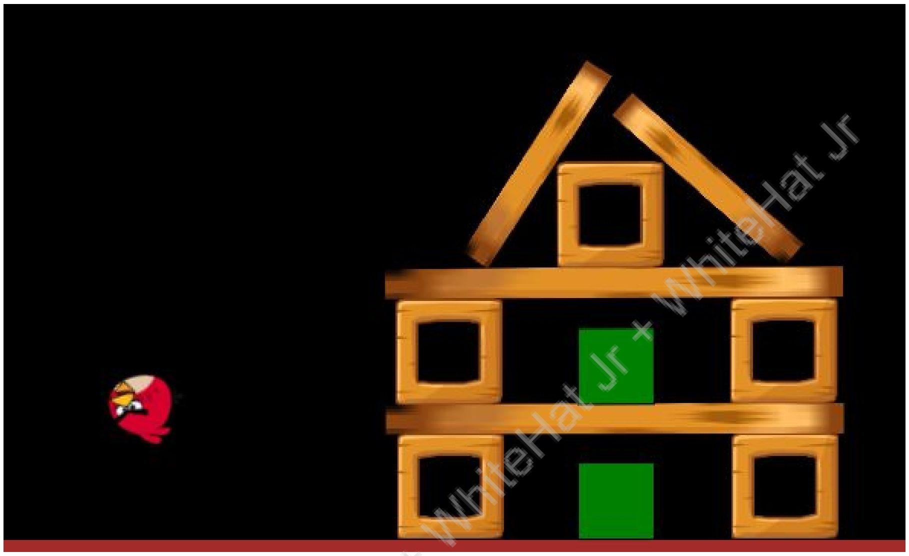
```
class Box extends BaseClass {
  constructor(x, y, width, height){
    super(x,y,width,height);
    this.image = loadImage("sprites/wood1.png");
  }

}
```

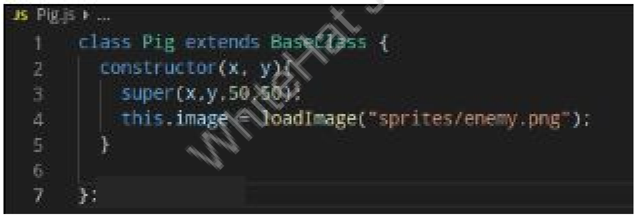


9. Inherit the log class and add images similar to other objects.

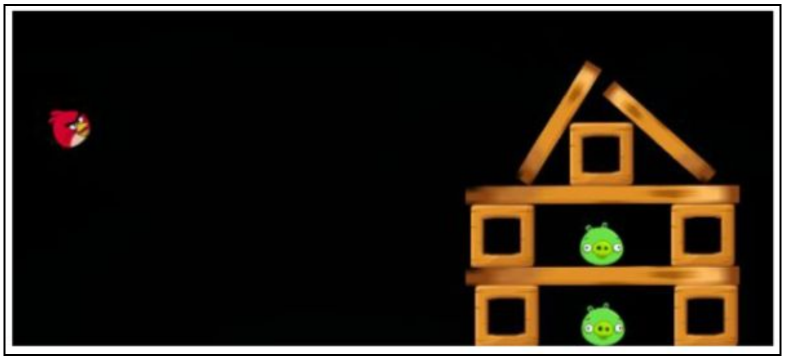


```
class Log extends BaseClass{
  constructor(x,y,height,angle){
    super(x,y,20,height,angle);
    this.image = loadImage("sprites/wood2.png");
    Matter.Body.setAngle(this.body, angle);
  }
}
```

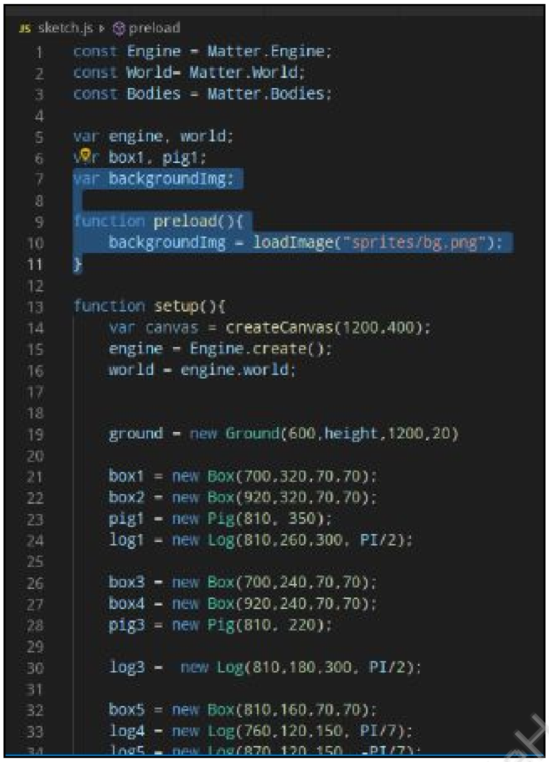




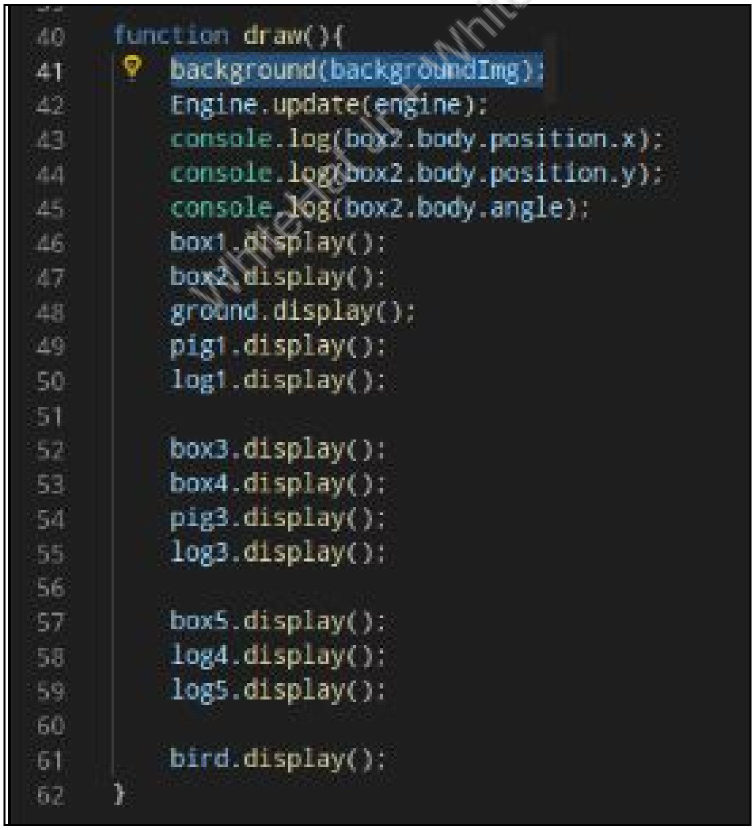
```
class Pig extends BaseClass {
  constructor(x, y){
    super(x,y,50,50);
    this.image = loadImage("sprites/enemy.png");
  }

};
```



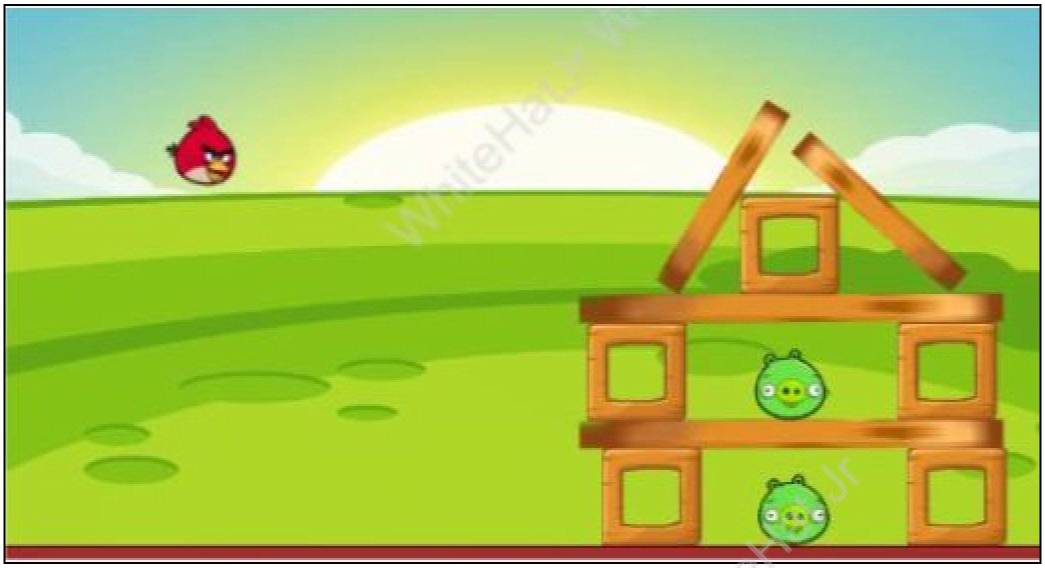
9. Add the background image in the sketch file



```
const Engine = Matter.Engine;
const World= Matter.World;
const Bodies = Matter.Bodies;

var engine, world;
var box1, pig1;
var backgroundImg;

function preload(){
    backgroundImg = loadImage("sprites/bg.png");
}

function setup(){
    var canvas = createCanvas(1200,400);
    engine = Engine.create();
    world = engine.world;


    ground = new Ground(600,height,1200,20)

    box1 = new Box(700,320,70,70);
    box2 = new Box(920,320,70,70);
    pig1 = new Pig(810, 350);
    log1 = new Log(810,260,300, PI/2);

    box3 = new Box(700,240,70,70);
    box4 = new Box(920,240,70,70);
    pig3 = new Pig(810, 220);

    log3 =  new Log(810,180,300, PI/2);

    box5 = new Box(810,160,70,70);
    log4 = new Log(760,120,150, PI/7);
    log5 = new Log(870,120,150, -PI/7);
```



```
function draw(){
    background(backgroundImg);
    Engine.update(engine);
    console.log(box2.body.position.x);
    console.log(box2.body.position.y);
    console.log(box2.body.angle);
    box1.display();
    box2.display();
    ground.display();
    pig1.display();
    log1.display();

    box3.display();
    box4.display();
    pig3.display();
    log3.display();

    box5.display();
    log4.display();
    log5.display();

    bird.display();
}
```



## What's NEXT?

In the next class, you will learn about Git and GitHub.

## EXTEND YOUR KNOWLEDGE:

1. Explore more examples of inheritance here: https://p5js.org/examples/objects-inheritance.html