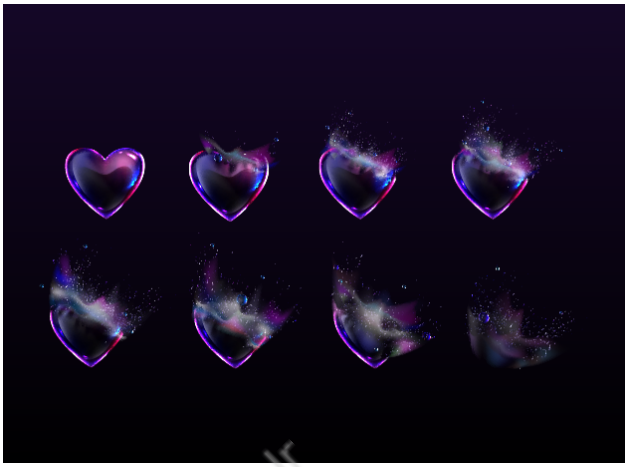
SPRITESHEET AND ANIMATION



What is our GOAL for this MODULE?

In this class, we learned to create **SpriteSheets** for animations and added animations to the boats.

What did we ACHIEVE in the class TODAY?

- Used the Spritesheet to add the sailing animation to the boat.
- Used the Spritesheet to add broken boat animation to the boat.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- **SpriteSheet**
- **setTimeout()**
- **this.speed**

How did we DO the activities?

1. Inspect the boat.json file.

2. In the boat.png file you see that we have 4 different frames of the boat, each has a different action.

3. Create a boatAnimation array and also declare two variables **boatSpritedata** and **boatSpriteSheet. boatSpritedata** contained data from JSON & **boatSpritesheet** contained the images.

```
var boatAnimation = [];
var boatSpritedata, boatSpritesheet;
```

4. Load Images and JSON in **preload()** function.

```
function preload() {
  backgroundImg = loadImage("./assets/background.gif");
  towerImage = loadImage("./assets/tower.png");
  boatSpritedata = loadJSON("assets/boat/boat.json");
  boatSpritesheet = loadImage("assets/boat/boat.png");

}
```

5. Write a code to iterate through an array of **boatFrames** in **setup()** function in **sketch.js** file.

```
var boatFrames = boatSpritedata.frames;
for (var i = 0; i < boatFrames.length; i++) {
  var pos = boatFrames[i].position;
  var img = boatSpritesheet.get(pos.x, pos.y, pos.w, pos.h);
  boatAnimation.push(img);
}
```

6. Write code in the **Boat.js** file in **constructor ();** Add the **boatAnimation** as a parameter to the **constructor** function and then using **this.animation** add it to the boat.

```
class Boat {
  constructor(x, y, width, height, boatPos, boatAnimation) {
    var options = {
      restitution: 0.8,
      friction: 1.0,
      density: 1.0,
      label: "boat"
    };
    this.animation = boatAnimation;
    this.speed = 0.05;
    this.body = Bodies.rectangle(x, y, width, height, options);
    this.width = width;
    this.height = height;

    this.boatPosition = boatPos;
    this.image = loadImage("assets/boat.png");
    World.add(world, this.body);
  }
```

7. Write code in the **Boat.js** file in the **display()** function.

```
display() {
  var pos = this.body.position;
  var index = floor(this.speed % this.animation.length);

  push();
  translate(pos.x, pos.y);
  imageMode(CENTER);
  image(this.animation[index], 0, this.boatPosition, this.width, this.height);
  pop();
}
```

8. Write code in **sketch.js** file in **showBoats** function.

```
function showBoats() {
  if (boats.length > 0) {
    if (
      boats[boats.length - 1] === undefined ||
      boats[boats.length - 1].body.position.x < width - 300
    ) {
      var positions = [-40, -60, -70, -20];
      var position = random(positions);
      var boat = new Boat(
        width,
        height - 100,
        170,
        170,
        position,
        boatAnimation
      );

      boats.push(boat);
    }

    for (var i = 0; i < boats.length; i++) {
      if (boats[i]) {
        Matter.Body.setVelocity(boats[i].body, {
          x: -0.9,
          y: 0
        });

        boats[i].display();
        boats[i].animate();

    }
    }
  } else {
    var boat = new Boat(width, height - 60, 170, 170, -60, boatAnimation);
    boats.push(boat);
  }
}
```

9. Give the name of the **image** and the **positions** of the **frames**.

10. Create an empty array called **brokenBoatAnimation** and load the data to it.

```
brokenBoatSpritedata = loadJSON("assets/boat/broken_boat.json");
brokenBoatSpritesheet = loadImage("assets/boat/broken_boat.png");
```

11. Use **SpriteSheet** to get the image with respective position and push this image to the **boatAnimation** array.

```
var brokenBoatFrames = brokenBoatSpritedata.frames;
for (var i = 0; i < brokenBoatFrames.length; i++) {
  var pos = brokenBoatFrames[i].position;
  var img = brokenBoatSpritesheet.get(pos.x, pos.y, pos.w, pos.h);
  brokenBoatAnimation.push(img);
}
```

12. Play the animation before the boat disappears and the boat fades away slowly.

```
remove(index) {
  this.animation = brokenBoatAnimation;
  this.speed = 0.05;
  this.width = 300;
  this.height = 300;
  this.isBroken = true;
  setTimeout(() => {
    Matter.World.remove(world, boats[index].body);
    delete boats[index];
  }, 2000);
}
```


What's next?

In the next class, we'll add sounds to the game and write conditions for game over and score.

EXTEND YOUR KNOWLEDGE

1. Bookmark the following link to know more about setTimeout().