

SOUNDS AND GAME OVER



What is our GOAL for this MODULE?

In this class, we've learned to add the sounds to the game along with the scoring and game over functionality.

What did we ACHIEVE in the class TODAY?

- Write the condition in the **showBoats()** function.
- Add a **sweet alert library** to the code.
- Write the code for the **gameOver()** function.
- Load the sound files in the **preload** function.
- Play the **background** sound in the **draw** function.
- Play **cannonExplosion** sound when we shoot the cannonball.
- Add sound when the **cannonball** hits the water.
- Add the **score** functionality to the game and display the score.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- Adding Sounds
- Game Over functionality
- Score

How did we DO the activities?

1. Write the condition in the **showBoats()** function inside the first **if** condition, just after where we pushed the boat in the boats array.

```

    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();
      var collision = Matter.SAT.collides(tower, boats[i].body);
    } else {
      boats[i];
    }
  }
} else {
  var boat = new Boat(width, height - 60, 170, 170, -60, boatAnimation);
  boats.push(boat);
}

```

2. Create a **this.isBroken** and set its value to false as initially the boat is not broken.

```

class Boat {
  constructor(x, y, width, height, boatPos, boatAnimation) {
    var options = {
      restitution: 0.8,
      friction: 1.0,
      density: 1.0
    };

    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.isBroken = false;

    World.add(world, this.body);
  }
  animate() {
    this.speed += 0.05 % 1.1;
  }
}

```

3. Add a **sweet alert library** to the code; to use sweet alert we call the **swal()** and inside the round brackets that creates an object with the information that we want to display.

```
<!-- Sweet Alert -->
<script
  src="https://code.jquery.com/jquery-3.5.1.min.js"
  integrity="sha256-9/aliU8dGd2tb60SsuzixeV4y/faTqgFtohetphbbj0="
  crossorigin="anonymous"
></script>
<script src="./lib/sweetalert.min.js"></script>
<link rel="stylesheet" type="text/css" href="./lib/sweetalert.css" />
<link rel="stylesheet" type="text/css" href="style.css" />
```

4. Write the code for the **gameOver()** function. When the game is over we wanted the user to see the game over message and have a reload button which started the game again for the player to play.

```
function gameOver() {
  swal(
    {
      title: `Game Over!!!`,
      text: "Thanks for playing!!",
      imageUrl:
        "https://raw.githubusercontent.com/whitehatjr/PiratesInvasion/main/assets/boat.png",
      imageSize: "150x150",
      confirmButtonText: "Play Again"
    },
    function(isConfirm) {
      if (isConfirm) {
        location.reload();
      }
    }
  );
}
```

5. Load the sound files in the **preload** function.

```
function preload() {  
  backgroundImg = loadImage("./assets/background.gif");  
  backgroundMusic = loadSound("./assets/background_music.mp3");  
  waterSound = loadSound("./assets/cannon_water.mp3");  
  pirateLaughSound = loadSound("./assets/pirate_laugh.mp3");  
  cannonExplosion = loadSound("./assets/cannon_explosion.mp3");  
}
```

6. Play the **background** sound in the **draw** function.

```
function draw() {  
  background(189);  
  image(backgroundImg, 0, 0, width, height);  
  
  if (!backgroundMusic.isPlaying()) {  
    backgroundMusic.play();  
    backgroundMusic.setVolume(0.1);  
  }  
}
```

7. Play **cannonExplosion** sound when we shoot the cannonball.

```
function keyReleased() {  
  if (keyCode === DOWN_ARROW) {  
    cannonExplosion.play();  
    balls[balls.length - 1].shoot();  
  }  
}
```

8. Add sound when the **cannonball** hits the water.

```
function showCannonBalls(ball, index) {
  ball.display();
  ball.animate();
  if (ball.body.position.x >= width || ball.body.position.y >= height - 50) {
    if (!ball.isSink) {
      waterSound.play();
      ball.remove(index);
    }
  }
}
```

9. Write the condition to check if the **isLaughing** flag is false and if the pirate laugh sound is playing.

```
var collision = Matter.SAT.collides(tower.body, boats[i].body);
if (collision.collided && !boats[i].isBroken) {
  //Added isLaughing flag and setting islaughing to true
  if(!isLaughing && !pirateLaughSound.isPlaying()){
    pirateLaughSound.play();
    isLaughing = true
  }
  isGameOver = true;
  gameOver();
}
```

10. Add the score functionality to the game, declaring a score variable and increase the score when the cannonball hits the boat and finally display the score.

```
for (var i = 0; i < balls.length; i++) {
  showCannonBalls(balls[i], i);
  for (var j = 0; j < boats.length; j++) {
    if (balls[i] !== undefined && boats[j] !== undefined) {
      var collision = Matter.SAT.collides(balls[i].body, boats[j].body);
      if (collision.collided) {
        if (!boats[j].isBroken && !balls[i].isSink) {
          score += 5;
          boats[j].remove(j);
          j--;
        }
      }
    }
  }
}
```

11. Display the score.

```
fill("#6d4c41");  
textSize(40);  
text(`Score:${score}`, width - 200, 50);  
textAlign(CENTER, CENTER);
```

What's next?

In the next class, we'll be working to create a new game "Bunny Catch Fruits" using the Physics Engine.

EXTEND YOUR KNOWLEDGE

1. Bookmark the following link to experiment and create more SpriteSheets at:
<https://www.piskelapp.com/>