

Arrays and Sound



What is our GOAL for this MODULE?

In this class, we learned to add sound effects and animation to the game.

What did we ACHIEVE in the class TODAY?

- Learned about arrays and their basic operations.
- Played some “hit” sound when the ball is bouncing off the paddle.
- Selected the sounds from the library.
- Added and displayed score to the game.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- Arrays and their basic operations.
- Add sound effects.
- Add animation life to the game.
- Add score to the game.

How did we DO the activities?

1. Arrays and their basic operations.

A. Store multiple names:

```
Workspace Version History Show Blocks
var name = "Alisha";
var friends = ['Adam', 'Parker', 'Virat', 'Kate', 'Tony'];
```

B. Store multiple type data:

```
Workspace Version History Show Blocks
var prime_numbers = [2,3,5,7,11]
var friends = ['Adam', 'Parker', 'Virat', 'Kate', 'Tony'];
var mix = ['Adam', 13, 'Sydney', 'Male'];
```

C. Access the first element of the array:

```
Workspace Version History Show Blocks
var friends = ['Adam', 'Parker', 'Virat', 'Kate', 'Tony'];
console.log(friends[0]);
```

```
Debug Console
"Adam"
```

2. Print alternate elements of the planet's array (array already defined in the program):

```

Workspace Version History Show Blocks
1 var planets = ['Mercury', 'Venus', 'Earth', 'Mars', 'Jupiter', 'Saturn',
2 'Uranus', 'Neptune'];
3
4 console.log(planets[0]);
5 console.log(planets[2]);
6 console.log(planets[4]);
7 console.log(planets[6]);
8

```

```

Debug Console Debug Spr
"Mercury"
"Earth"
"Jupiter"
"Uranus"

```

3. Add a code to destroy the brick when the ball bounces off it, and add customization in the **bounceOff()** function. Also, create a **brickHit()** function and call **brick.destroy()** in **brickHit()** function.

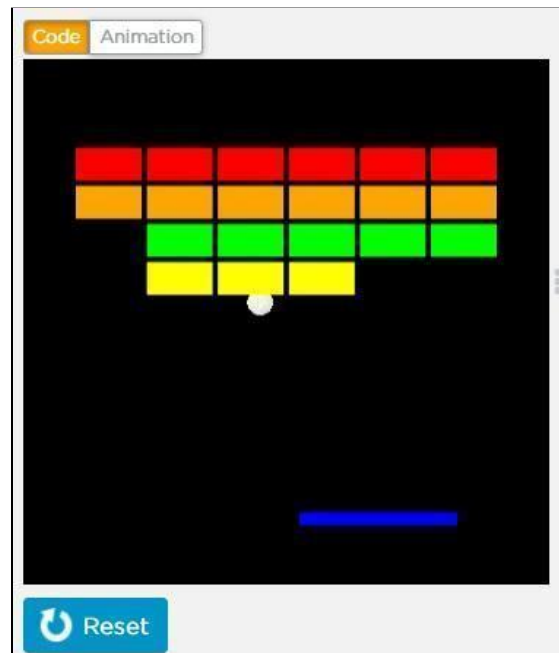
CODE:

```

31 function draw(){
32   background("black");
33
34   paddle.x = World.mouseX;
35
36   if(paddle.x < 60){
37     paddle.x =60;
38   }
39
40   if(paddle.x > 340){
41     paddle.x =340;
42   }
43   drawSprites();
44   ball.bounceOff(topEdge);
45   ball.bounceOff(leftEdge);
46   ball.bounceOff(rightEdge);
47   ball.bounceOff(paddle);
48   ball.bounceOff(bricks, brickHit);
49 }
50
46 ball.bounceOff(rightEdge);
47 ball.bounceOff(paddle);
48 ball.bounceOff(bricks, brickHit);
49 }
50
51 function mousePressed(){
52   ball.velocityX = 4;
53   ball.velocityY = 2;
54 }
55
56 function brickHit(ball, brick) {
57   brick.destroy();
58 }
59
60

```

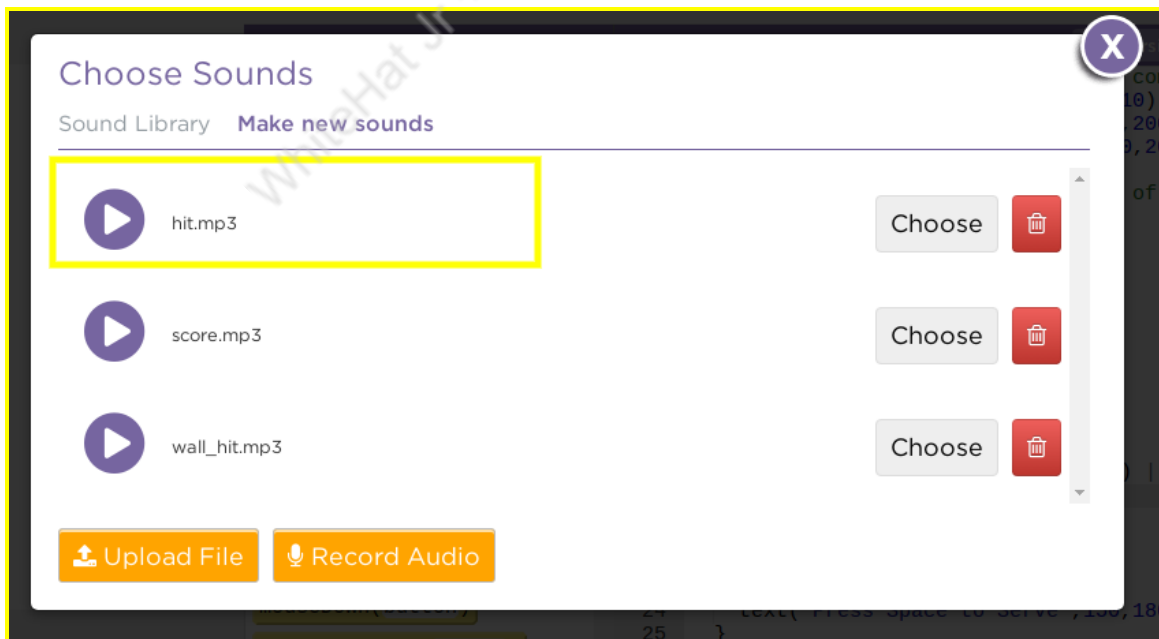
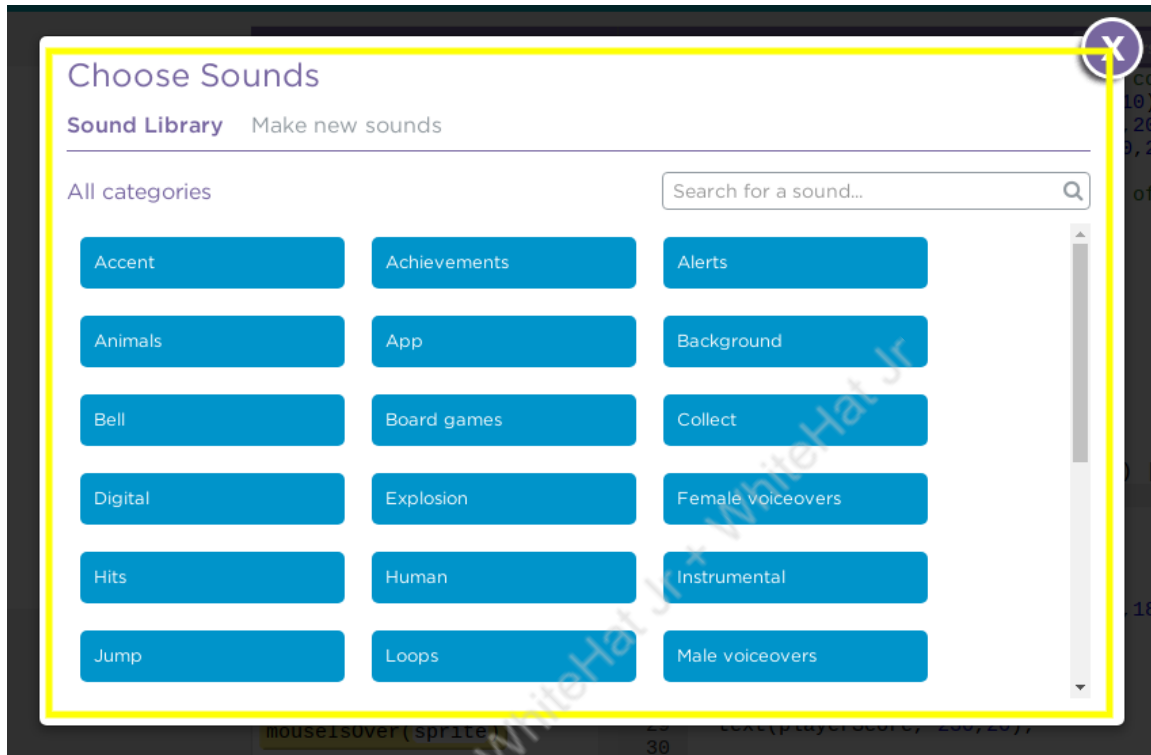
OUTPUT:



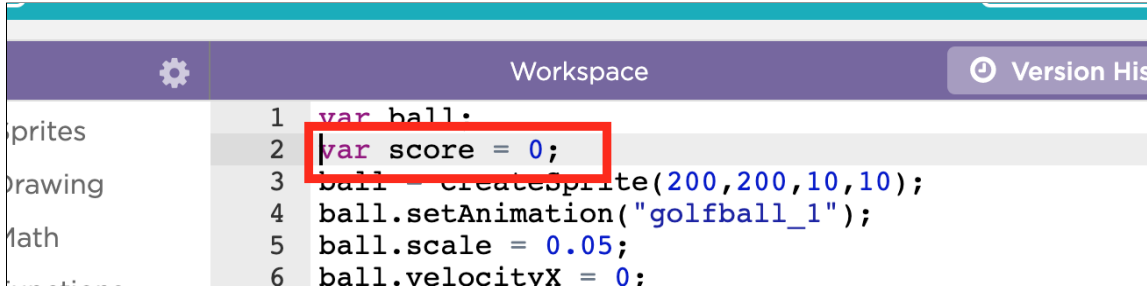
4. Add a "hit" sound if the paddles hit the ball.

```
ball.bounceOff(bricks, brickHit  
  
if(ball.bounceOff(paddle))  
{  
  playSound("hit")  
}
```

5. Choose sounds from the available **Sound Library** OR create new sounds by uploading a file or recording some sound.



6. Add score to the game. To display the score on the screen, it has to be stored somewhere.



```
1 var ball = {
2   var score = 0;
3   createSprite(200,200,10,10);
4   ball.setAnimation("golfball_1");
5   ball.scale = 0.05;
6   ball.velocityX = 0;
```

```
function brickHit(ball, brick) {
  brick.remove();
  score = score+5;
}
```

7. Display the score on the game canvas. Decide a position in such a way that the Score doesn't hamper the game visibility of the player.

```
33
34 function draw() {
35   background("black");
36
37   textSize(20);
38   text("Score: "+score,40,25);
39
40   paddle.x = World.mouseX;
41
42   if(paddle.x < 60)
```

8. Choose the sound and animation effects for the soccer practice game.

```
function brickHit(ball, brick) {  
  playSound("sound://category hits/puzzle game button 04.mp3")  
  brick.remove();  
  score = score+5;  
}
```

```
51 drawSprites();  
52 //rotation = rotation + 5;  
53 ball.bounceOff(topEdge);  
54 ball.bounceOff(leftEdge);  
55 ball.bounceOff(rightEdge);  
56 //ball.bounceOff(paddle);  
57 ball.bounceOff(bricks, brickHit);  
58 if(ball.bounceOff(paddle))  
59 {  
60   playSound("sound://category_tap/pu  
61 }  
62 if(!bricks[0])  
63 {  
64   //console.log("Won");  
65   ball.velocityX = 0;  
66   ball.velocityY = 0;  
67   text("Well Done!!",150,200);  
68 }  
69 }  
70
```

What's next?

In the next class, we will learn the very important concept of GAME STATE (mode).

Extend Your Knowledge:

Bookmark following link: it will be a reference for stopSound()

<https://studio.code.org/docs/gamelab/stopSound/>