



## Matrices y sonido

### ¿Cuál es nuestro objetivo para este módulo?

En esta clase, aprendimos a agregar efectos de sonido y animación al juego.

### ¿Qué logramos en clase el día de hoy?

Aprendimos sobre matrices y sus operaciones básicas.

Reproducimos un sonido de "golpe" cuando la pelota rebota en la paleta.

Seleccionamos los sonidos de la biblioteca.

Agregamos la puntuación y mostramos el puntaje en el lienzo del juego.

### ¿Qué conceptos/bloques de código cubrimos el día de hoy?

Matrices y sus operaciones básicas.

Agregar efectos de sonido.

Agregar animation life (animación) al juego.

Añadir la puntuación al juego.

### ¿Cómo hicimos las actividades?

1. Matrices y sus operaciones básicas.

A. Almacenar varios nombres:

```

Espacio de trabajo:
1 var name = "Alisha";
2
3 var friends = ['Adam', 'Parker', 'Virat', 'Kate', 'Tony'];
```

B. Almacenar datos de varios tipos:

```

Espacio de trabajo: Historial de versiones
var prime_numbers = [2, 3, 5, 7, 11];

var name = "Alisha";

var friends = ['Adam', 'Parker', 'Virat', 'Kate', 'Tony'];
```

C. Acceder al primer elemento de la matriz:

```

Espacio de trabajo: Historial de versiones
var friends = ['Adam', 'Parker', 'Virat', 'Kate', 'Tony'];
console.log(friends[0]);
```

```

Consola
"Adam"
```

D. Agregar elementos a la matriz usando la función **push()** (insertar):

```

Espacio de trabajo: Historial de versiones
1  var friends = ['Adam', 'Parker', 'Virat', 'Kate', 'Tony'];
2  console.log(friends);
3
4  friends.push('Michael');
5  console.log(friends);

```

```

Consola
▼ ["Adam", "Parker", "Virat", "Kate", "Tony"]
  0: "Adam"
  1: "Parker"
  2: "Virat"
  3: "Kate"
  4: "Tony"
▼ ["Adam", "Parker", "Virat", "Kate", "Tony", "Michael"]
  0: "Adam"
  1: "Parker"
  2: "Virat"
  3: "Kate"
  4: "Tony"
  5: "Michael"

```

E. Eliminar el último elemento de una matriz usando la función **pop()** (eliminar):

```

Espacio de trabajo: Historial de versiones
1  var friends = ['Adam', 'Parker', 'Virat', 'Kate', 'Tony', 'Michael'];
2  console.log(friends);
3
4  friends.pop();
5  friends.pop();
6  console.log(friends);

```

```
Consola
▼ ["Adam", "Parker", "Virat", "Kate", "Tony", "Michael"]
  0: "Adam"
  1: "Parker"
  2: "Virat"
  3: "Kate"
  4: "Tony"
  5: "Michael"
▼ ["Adam", "Parker", "Virat", "Kate"]
  0: "Adam"
  1: "Parker"
  2: "Virat"
  3: "Kate"
```

- Imprimir elementos alternativos de la matriz planets (planetas) (matriz ya definida en el programa):

Espacio de trabajo: Historial de versiones

```

1 var planets = ['Mercurio', 'Venus', 'Tierra', 'Marte', 'Júpiter',
2               'Saturno', 'Urano', 'Neptuno'];
3
4 console.log(planets[0]);
5 console.log(planets[2]);
6 console.log(planets[4]);
7 console.log(planets[6]);

```

Consola

```

"Mercurio"
"Tierra"
"Júpiter"
"Urano"

```

- Agregar un código para destruir el ladrillo cuando la pelota rebote y agregar personalización en la función **bounceOff()** (rebotar en). Además, crear una función **brickHit()** (golpear ladrillo) y llamar a **brick.destroy()** (destruir ladrillo) en la función **brickHit()**.

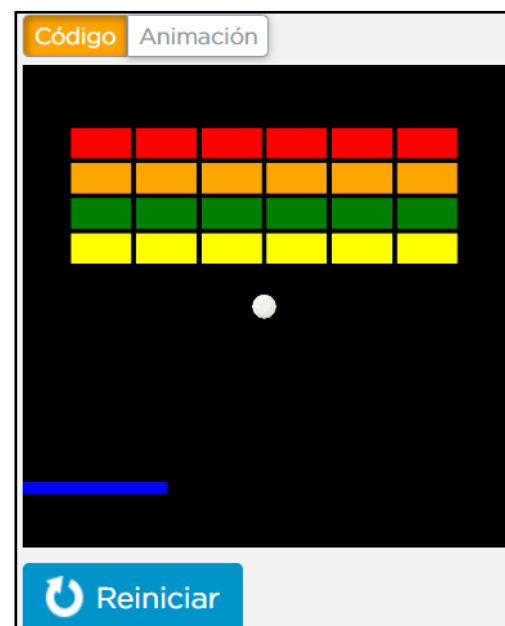
### CÓDIGO:

```

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

```

### OUTPUT:



```
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
```

4. Agregar un sonido de "golpe" si las paletas golpean la pelota.

```
ball.bounceOff(bricks, brickHit);

if(ball.bounceOff(paddle))
{
  playSound("|")
}
```

5. Elegir sonidos de la **biblioteca de sonidos** disponibles o crear nuevos sonidos cargando un archivo o grabando algún sonido.

## Elegir sonidos

Biblioteca de sonidos [Crear nuevos sonidos](#)

All categories

🔍 Buscar un sonido...

Accent

Achievements

Alerts

Animals

App

Background

Bell

Board games

Collect

Digital

Explosion

Female voiceovers

Hits

Human

Instrumental

Jump

Loops

Male voiceovers

## Elegir sonidos

Biblioteca de sonidos [Crear nuevos sonidos](#)

All categories

🔍 hit

▶ vibrant\_game\_correct\_...  
00:01

▶ vibrant\_wrong\_action\_...  
00:01

▶ app\_button\_1.mp3  
Less than 1 second

▶ app\_button\_2.mp3  
Less than 1 second

▶ app\_button\_3.mp3  
Less than 1 second

▶ app\_button\_4.mp3  
Less than 1 second

▶ app\_button\_5.mp3  
Less than 1 second

▶ app\_button\_6.mp3  
Less than 1 second

▶ app\_button\_click\_1.mp3  
Less than 1 second

▶ app\_button\_click\_2.mp3  
Less than 1 second

▶ app\_button\_slide\_cool...  
Less than 1 second

▶ app\_button\_slide\_cool...  
Less than 1 second

▶ app\_ice\_button\_click\_1...  
Less than 1 second

▶ app\_ice\_button\_click\_...  
Less than 1 second

▶ app\_ice\_button\_click\_...  
Less than 1 second

▶ app\_interface\_button\_...

▶ app\_interface\_button\_...

▶ app\_interface\_button\_...

Choose



## Elegir sonidos

Biblioteca de sonidos [Crear nuevos sonidos](#)

Go to the "Sound library" to find sounds for your project.  
To upload your own sound, click "Upload File." Your uploaded assets will appear here.

 Subir archivo

 Grabar audio

6. Añadir puntuación al juego. Para mostrar la puntuación en la pantalla, debe almacenarse en algún lugar.

### Espacio de trabajo:

```
1 var ball;
2 var score = 0;
3 ball = 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. Mostrar la puntuación en el lienzo del juego. Decidir una posición de tal manera que la puntuación no obstaculice la visibilidad del jugador en el juego.

```
function draw() {
  background("black");

  text("Puntuación: "+score, 40, 25);

  paddle.x = World.mouseX;
}
```

8. Elegir los efectos de sonido y animación para el juego de práctica de fútbol.

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

```
drawSprites();  
//rotación = rotation + 5;  
ball.bounceOff(topEdge);  
ball.bounceOff(leftEdge);  
ball.bounceOff(rightEdge);  
//ball.bounceOff(paddle);  
ball.bounceOff(bricks, brickHit);  
if(ball.bounceOff(paddle))  
{  
  playSound("sound://category_tap/puzzle_game_organic_wood_block_tone_tap_1.mp3");  
}  
  
if(!bricks[0])  
{  
  //console.log("Won");  
  ball.velocityX =0;  
  ball.velocityY =0;  
  text(";Bien hecho!",150,200);  
}  
}
```

### ¿Qué sigue?

En la siguiente clase aprenderemos un concepto muy importante: GAME STATE(mode) (Estado del Juego).

### Amplía tu conocimiento:

Guarda el siguiente enlace: será una referencia para stopSound() (detener el sonido)

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