



ESTADOS DEL JUEGO

¿Cuál es nuestro objetivo este módulo?

En esta clase aprendimos a almacenar el estado (modo) de un juego en una variable, y a asignar un comportamiento diferente a los objetos en el juego, dependiendo del estado del juego.

¿Qué logramos en clase el día de hoy?

Agregamos una instrucción **text()** (texto), cerca del centro que dice "Haz clic para sacar la pelota".

Creamos una variable llamada **gamestate** (modo) o estado del juego, y le dimos un estado inicial (modo) de "serve".

Cambiamos el estado del juego (modo), después de que el usuario hiciera clic en un botón del mouse.

Programamos el tercer estado del juego, es decir, el modo Game Over (fin del juego).

Detectamos una pulsación de tecla en la barra espaciadora que cambió el estado del juego entre play(jugar) y pause (pausa).

Mostramos algunas instrucciones de texto más, para que el jugador sepa si es un estado pause o game over.

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

Uso de programación condicional.

Operadores lógicos.

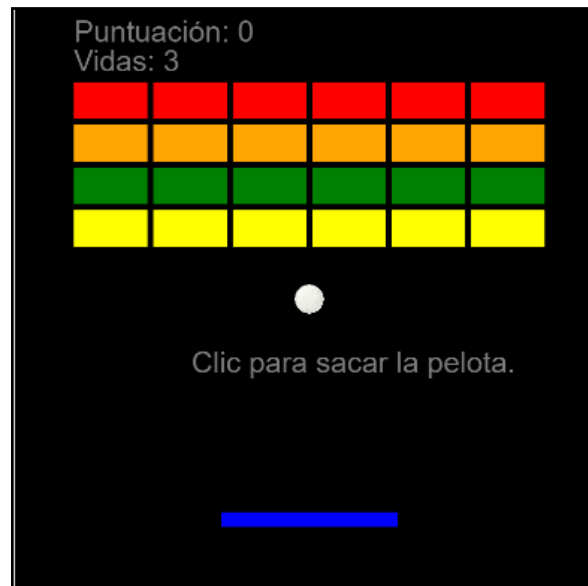
Agregar instrucciones `text()` (texto) en la consola.

Cómo agregar estados del juego en el juego

¿Cómo hicimos las actividades?

1. Agregar una instrucción **text()** cerca del centro, que diga "Haz clic para sacar la pelota":

```
function draw() {  
  background("black");  
  textSize(20);  
  text("Click to serve the ball.", 120, 250);  
  text("Score: "+score, 40, 25);
```



2. Crear una variable llamada gamestate (mode) , lo que significa estado del juego (modo), y darle un estado inicial (modo) "**serve**". Siempre que almacenamos texto dentro de una variable lo ponemos en doble comillas (" ").

```
1 var ball;
2 var score = 0;
3 var gamestate = "serve";
4 ball = createSprite(200,200,10,10);
5 ball.setAnimation("golfball_1");
6 ball.scale = 0.05;
7 var paddle = createSprite(200, 350, 120, 10);
8 paddle.shapeColor = color(0,0,255);
9
10 createEdgeSprites();
11 var colors = [color(255,0,0),color(255,165,0), color(0,255,0),color(0,0,255)];
12 var BRICK_W = 50;
13 var BRICK_H = 25;
14 var BRICK_MARGIN = 4;
15
16 var offsetY = 80;
```

3. Cambiar el estado del juego (modo) después de que el usuario haga clic en un botón del mouse, y cambiar la variable gameState a **"play"** (jugar) después de pulsar la tecla de espacio. ¡Recuerda que los valores dentro de las variables pueden cambiar! Por eso se llaman "variables":

```
}
function mousePressed()
{
    if(gamestate == "serve")
    {
        gamestate = "play";
        ball.velocityY = -7;
        ball.velocityX = 7;
    }
}
```

4. Escribir un código para detectar un toque entre la pelota y el borde inferior del lienzo.

```
62 }
63 if(!bricks[0])
64 {
65     //console.log("Won");
66     ball.velocityX = 0;
67     ball.velocityY = 0;
68     text("Well Done!!",150,200);
69 }
70 if(ball.isTouching(bottomEdge)) {
71     lifeover();
72 }
73
74 }
75
76 function mousePressed()
77 {
```

5. Crear una función personalizada **lifeover()** (vidas terminadas) para reducir la cantidad de vidas en 1 cuando un jugador es golpeado por un enemigo, también verifica cuántas vidas quedan y, en consecuencia, establece el estado del juego.

```
3
4 function lifeover()
5 {
6     lives = lives-1;
7
8     if(lives >= 1)
9     {
10         gamestate = "serve";
11     }
12     else
13     {
14         gamestate = "end";
15     }
16 }
17
```

6. Mostrar otras instrucciones de texto para que el jugador sepa si es un estado serve o un estado game over.

```
28
29
30
31 function draw() {
32   background("black");
33   textSize(20);
34   text("Score: "+score,40,25);
35   text("Lives:"+lives, 50, 50);
36   if(gamestate == "serve")
37   {
38     text("Click to serve the ball.", 120, 250);
39     ball.velocityX = 0;
40     ball.velocityY = 0;
41     ball.x = 200;
42     ball.y = 200;
43   }
44   else if(gamestate == "end")
45   {
46     text("Game Over", 150, 250);
47     ball.remove();
48   }
49   else
50   {
51     gameplay();
52   }
53
54   drawSprites();
55 }
56
57
```

¿Qué sigue?

En la siguiente clase aprenderemos cómo aumentar la complejidad del juego y aplicar inteligencia artificial al juego.

Amplía tu conocimiento:

Guarda el siguiente enlace: será una referencia para el operador de Desigualdad.

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