```
//html
<!DOCTYPE html>

<html>
<head>
  <meta http-equiv="CONTENT-TYPE"
content="text/html; charset=UTF-8">

    <link rel="stylesheet"
href="index.css">
</head>
<body>

  <div class="box">
    <p id="score">0</p>
    <p id="mess">Game Over!</p>

     <div id="ball"></div>
  <div id="slide"></div>
  </div>
    <script src="index.js"></script>
</body>
</html>
//Css
body{
  margin:0px;
  padding:0px;
  background-color:pink;
}
#ball{
  position:absolute;
  height:10vh;
  width:10vh;
  border-radius:50%;


   background-color:Red;
   border:2px solid black;


  top:10px;
  background-image:url('Pictures/t.jpg');
  background-size:cover;



}
#score{
  position:absolute;
  font-size:15vh;
  text-align:center;
  top:20vh;


}
#mess{
  position:absolute;
  top:43vh;
  font-size:10vh;
  text-align:center;
  visibility:hidden;
}
.box{
  display:flex;

  justify-content:center;
  height:100vh;
  width:99vw;
  touch-action:none;
  margin-top:-17px;



}
#slide{
  position:absolute;
  height:10vh;
  width:25vw;
  background-color:yellow;
  border-radius:3rem;
  touch-action:none;
  border:2px solid black;
  top:89%;

}
//Javascript
const draggable = document.querySelector(”#slide”);
const score=document.querySelector(”#score”);
const mess=document.querySelector(”#mess”);
```

```
const
body=document.querySelector(”body”
);
const
ball=document.querySelector(”#ball”)
;
const
box=document.querySelector(”.box”);
const slideRect =
draggable.getBoundingClientRect();
const boxRect =
box.getBoundingClientRect();

let startX = 0; // Initial touch position
let currentX = 0; // Current position
of the element
let offsetX = 0; // Offset between
start and current touch points
```

```
let offsetY = 0;
let ballRadius = 25; // Radius of the
ball (half of 50px)
let ballSpeedX =
boxRect.width*(1/100); // Ball's
horizontal speed (pixels per frame)
let ballSpeedY =
boxRect.height*(0.7/100); // Ball's
vertical speed (pixels per frame)

let count=0;


 const colors =
["Green","Blue","Orange",
"Purple","Pink",    "Brown",
"White","Gray",    "Cyan","Magenta",
  "Lime","Olive",    "Navy" ];
let i=0;
```

```
function vibrate(){
     if(navigator.vibrate){
       navigator.vibrate(200);
     }
  }
function checkCollision() {
  // Get the bounding rectangles of
both elements
   // Check if they overlap
  const ballRect =
ball.getBoundingClientRect();
  const slideRect =
draggable.getBoundingClientRect();
  const boxRect =
box.getBoundingClientRect();



  let newX = ballRect.left +
ballSpeedX;
  let newY = ballRect.top + ballSpeedY;

  // Check for collision with the left or
right boundary of the box
  if (newX <= boxRect.left) {
    ballSpeedX =
Math.abs(ballSpeedX);
    vibrate();
    // Reverse direction to the right
  } else if (newX + ball.offsetWidth >=
boxRect.right) {
    ballSpeedX = -
Math.abs(ballSpeedX); // Reverse
direction to the left
    vibrate();
  }
  if (newY <= boxRect.top) {
    ballSpeedY =
Math.abs(ballSpeedY); // Reverse
direction downwards
    vibrate();
  }
  else if (newY + ball.offsetHeight >=
boxRect.bottom) {
    ballSpeedY = -
Math.abs(ballSpeedY);
    vibrate();
    ballSpeedX=0;
    ballSpeedY=0;

draggable.removeEventListener('touc
hmove',handleMove);
    mess.style.visibility='visible';
    return;
  }
  if (
    newX + ball.offsetWidth > slideRect.left &&
    newX < slideRect.right &&
    newY + ball.offsetHeight > slideRect.top &&
    newY < slideRect.bottom
  ) {
    // Ball has collided with the slide, reverse direction
    if (ballSpeedX > 0) {
      ballSpeedX = -Math.abs(ballSpeedX);// Ball hit the right side of the slide
      vibrate();
    } else {
      ballSpeedX = -
Math.abs(ballSpeedX);
      // Ball hit the left side of the slide
      vibrate();

    }
    if (ballSpeedY > 0) {
      ballSpeedY = -
Math.abs(ballSpeedY);
      vibrate();



      // Ball
    } else {
      ballSpeedY = -
Math.abs(ballSpeedY); // Ball hit the
```

```
top of the slide
     vibrate();




   }
  count+=1;
   ballSpeedX *= 1.10; // Increase
speed by 10%
   ballSpeedY *= 1.10; // Increase
speed by 10%

body.style.backgroundColor=colors[i];
    i+=1;
    if (i==15) i=0;


  }
  // Update the ball's position
  ball.style.left = `${newX - boxRect.left}px`; // Set left relative to the box
  ball.style.top = `${newY - boxRect.top}px`; // Set top relative to the box
  score.innerText=`${count}`;
}



setInterval(checkCollision, 50); // Animate ball every 50ms

 // Start touch
draggable.addEventListener('touchstart', (e) => {
  const touch = e.touches[0];
  offsetX = touch.clientX - slideRect.left;
  offsetY=touch.clientY-slideRect.top;
});

// Handle movement
function handleMove(e){
  const touch = e.touches[0];
  let new_X=touch.clientX-offsetX;
  let new_Y=touch.clientY-offsetY;
  if(new_X<boxRect.left){
    new_X=boxRect.left;
  }

if(new_X+slideRect.width>boxRect.right){
    new_X=boxRect.right-slideRect.width;
  }


draggable.style.left=new_X+boxRect.left+'px';

}
draggable.addEventListener('touchmove', handleMove);

// End touch
draggable.addEventListener('touchend', () => {
  // Update current position
  currentX = offsetX;
});
```