

Book Series
of
jago{coding}
Learn & Share {Your Code}

www.jagocoding.com

Deteksi Tubrukan Objek Dengan HTML5

[Part 1]

Oleh: Yudhistira Bayu W

Bismillahirrahmanirahim Assalamualaikum wr.wb Alhamdulillah kita telah sampai pada tutorial penanganan Tubrukan dengan menggunakan HTML5. Di tutorial ini akan terbagi 3 part, dimana part 1 berisi tentang konsep dasar, part 2 berisi tubrukan 2 objek, dan part 3 berisi konsep tubrukan banyak objek.

**Bismillahirrahmanirahim
Assalamualaikum wr.wb**

Alhamdulillah kita telah sampai pada tutorial penanganan Tubrukan dengan menggunakan HTML5. Di tutorial ini akan terbagi 3 part, dimana part 1 berisi tentang konsep dasar, part 2 berisi tubrukan 2 objek, dan part 3 berisi konsep tubrukan banyak objek.

Sebelumnya, kenapa sih tubrukan ini sangat penting? Ya wong memang di semua game menggunakan fitur ini kok :D Contohnya kaya bola dengan bola, pesawat dengan pesawat, dan lain sebagainya. Yang dibutuhin apa aja di konsep awal ini? Yang pasti kita menggunakan objek 2 bola. Berarti kita harus melakukan looping terhadap keduanya, dan menggambarkan bola berdasar posisi dan warna nya.

Oke langsung aja ke codingnya, seperti biasa; keterangan berada di akhir program ini.

```
<!DOCTYPE HTML>
<html>
  <head>
    <style>
      canvas {
        border-width: 1px;
        border-style: solid;
      }
    </style>
    <script>

      var canvas;
      var ctx;
      var balls = [];
      var circleX = 30;
      var circleY = 30;
      var circleXModifier = 2;
      var circleYModifier = 2;

      window.onload = function(){
```

```

canvas = document.getElementById("myCanvas");
ctx = canvas.getContext("2d");

    var newball1 = new Object();
    var newball2 = new Object();
    newball1.x = 30;
    newball1.y = 30;
    newball2.x = 500;
    newball2.y = 30;
    newball1.modx = 2;
    newball1.mody = 2;
    newball2.modx = -2;
    newball2.mody = 2;
    newball1.color = "#FF0000";
    newball2.color = "#0000FF";

    balls[0] = newball1;
    balls[1] = newball2;

    setInterval(draw, 1.6);
};

function draw() {
    ctx.fillStyle = "#fff";
    ctx.fillRect(0,0,canvas.width,canvas.height);

    for (b in balls) {
        ctx.beginPath();
        ctx.arc(balls[b].x, balls[b].y, 30, 0, 2 * Math.PI,
false);

        ctx.fillStyle = balls[b].color;
        ctx.fill();

        balls[b].x += balls[b].modx;
        balls[b].y += balls[b].mody;

        if (balls[b].x >= canvas.width-30)
            balls[b].modx = -2;
        else if (balls[b].x < 30)
            balls[b].modx = 2;

        if (balls[b].y >= canvas.height-30)
            balls[b].mody = -2;
        else if (balls[b].y < 30)
            balls[b].mody = 2;

    }

}
</script>
</head>

```

```
<body>
  <center>
    <canvas id="myCanvas" width="640" height="480">
      </canvas>
    </center>
  </body>
</html>
```

Keterangan:

- Kode di bawah ini, terdapat pada baris 14 yang berfungsi untuk membuat array balls yang dapat menampung banyak objek bola

```
var balls = [];
```

- Kode dibawah ini, kode yang terdapat pada baris 25 dan 26 yang berfungsi untuk membuat 2 objek bola

```
var newball1 = new Object();
var newball2 = new Object();
```

- Kode dibawah ini, terdapat pada baris 27-36 yang berfungsi untuk menentukan posisi x, y, modx, mody dan warna untuk kedua bola.

```
newball1.x = 30;
newball1.y = 30;
newball2.x = 500;
newball2.y = 30;
newball1.modx = 2;
newball1.mody = 2;
newball2.modx = -2;
newball2.mody = 2;
newball1.color = "#FF0000";
newball2.color = "#0000FF";
```

- Kode dibawah ini, terdapat pada baris 38 39 yang berfungsi untuk memasukan kedua bola tersebut kedalam array balls

```
balls[0] = newball1;
balls[1] = newball2;
```

- Kode dibawah adalah kode dari baris 48-67 yang berfungsi untuk melakukan loop array balls untuk kemudia bola di draw satu per satu. Dimana baris 49-53 itu yang berfungsi untuk men-draw atau enggambar bola sesuai posisi dan warnanya. Baris 55-56 dibawah ini juga berfungsi untuk mengubah setip bola dengan modx dan mody nya. Sementara untuk mengecek posisi x,y bola dan memberikan limit pada setiap tepi canvas terdapat pada baris 58-66. Fungsi dari memberikan limit pada tepi canvas adalah agar bola tetap berada di dalam canvas, tidak sampai keluar dari area tersebut.

```
for (b in balls) {
    ctx.beginPath();
    ctx.arc(balls[b].x, balls[b].y, 30, 0, 2 * Math.PI, false);
    ctx.fillStyle = balls[b].color;
    ctx.fill();

    balls[b].x += balls[b].modx;
    balls[b].y += balls[b].mody;

    if (balls[b].x >= canvas.width-30)
        balls[b].modx = -2;
    else if (balls[b].x < 30)
        balls[b].modx = 2;

    if (balls[b].y >= canvas.height-30)
        balls[b].mody = -2;
    else if (balls[b].y < 30)
        balls[b].mody = 2;
}
```

Jika anda telah mencoba program diatas, akan terlihat 2 buah bola dengan warna yang berbeda, bergerak secara acak dengan arah masing masing. Nah karena ini konsep dasar, jadi belum ada tubrukannya :D kapan tubrukannya? Nanti di part 2 dan part 3. Disini hanya membahas konsep dasarnya saja :D

Oke, cukup sekian yang dapat saya sampaikan. Bila ada pertanyaan silahkan ditanyakan :) Artikel ini akan berlanjut ke part 2 yang berisi tubrukan 2 buah objek.

Tentang Penulis



Yudhistira Bayu W

Seorang pelajar tampan dan rupawan yang mencoba belajar coding.