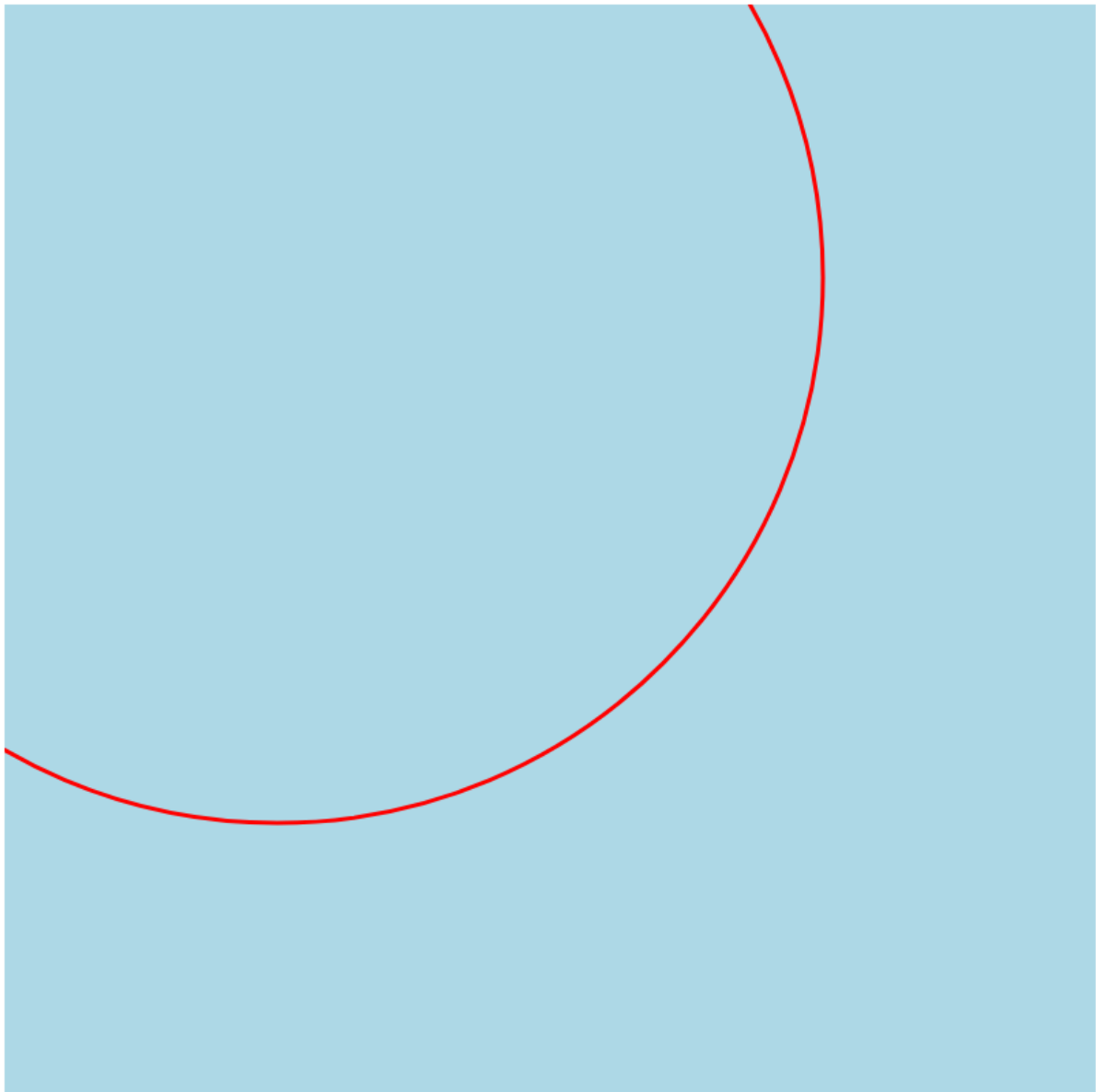


Laboratorijske vježbe iz predmeta Vizualizacija Podataka

Zadatak 1



Ovakav oblik dobijemo sljedećim kodom:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Z3-1</title>
  <script src="https://d3js.org/d3.v5.min.js"></script>
</head>

<body>
  <script>
    var svg = d3.select("body")
```

```

        .append("svg")
        .attr("width", "800")
        .attr("height", "800")
        .style("background-color", "lightblue");

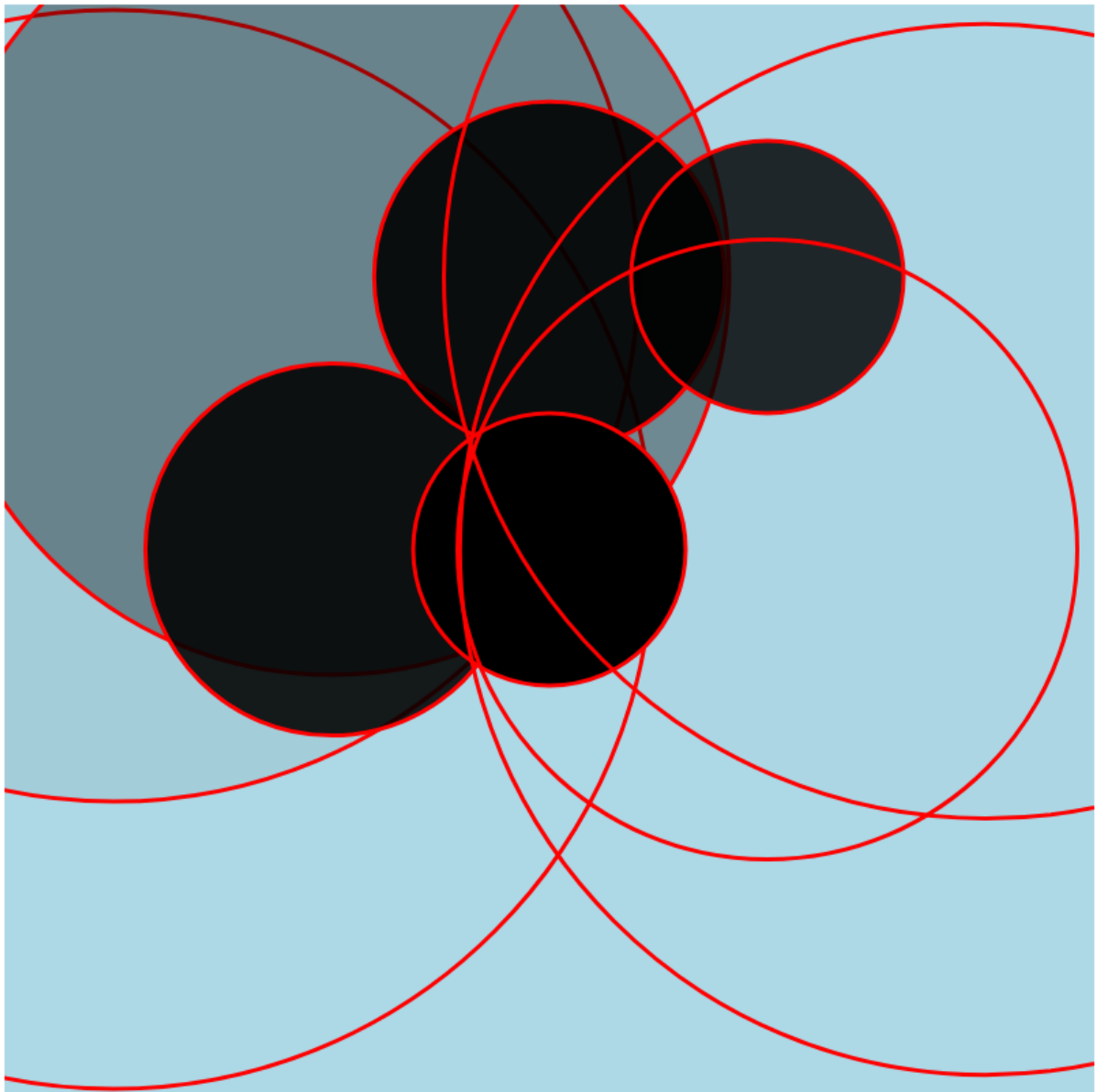
var circle = svg.append("circle")
    .attr("cx", "200")
    .attr("cy", "200")
    .attr("r", "100")
    .style("fill-opacity", "0")
    .style("stroke", "red")
    .style("stroke-width", "3");

circle.transition()
    .duration(3000)
    .style("fill-opacity", "0")
    .transition()
    .duration(5000)
    .style("fill-opacity", "1")
    .transition()
    .duration(3000)
    .attr("r", svg.attr("width") / 2)
    .style("fill-opacity", "0");
</script>
</body>
</html>

```

Napravimo krug te koristeći `transition()` animacije animiramo ga po želji. U ovom slučaju mu kažemo da čeka 3 sekunde, preko 5 sekundi da dobije ispunu te da se poveća ovisno o pripadajućem svg elementu i da nestane ispunu preko 3 sekunde.

Zadatak 2.



Kod:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Z3-2</title>
  <script src="https://d3js.org/d3.v5.min.js"></script>
</head>
<body>
  <script>
    var width = 800;
    var svg = d3.select("body")
      .append("svg")
```

```

        .attr("width",width)
        .attr("height","800")
        .style("background-color","lightblue");

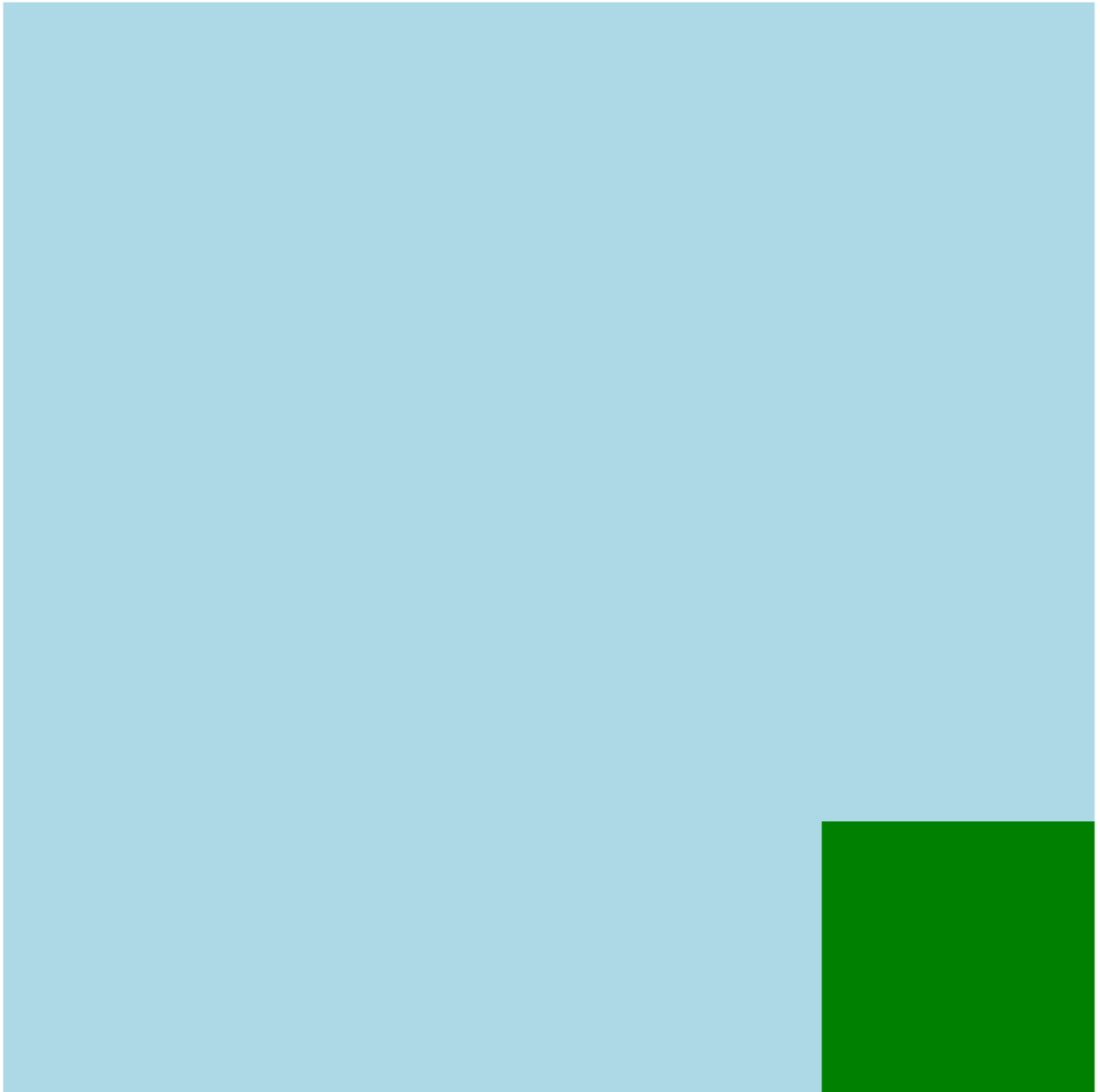
function moveCircle(circle){
    circle.transition()
        .duration(Math.ceil((Math.random()*3+2)*1000))
        .style("fill-opacity","0")
        .transition()
        .duration(Math.ceil((Math.random()*3+2)*1000))
        .style("fill-opacity","1")
        .transition()
        .duration(Math.ceil((Math.random()*3+2)*1000))
        .attr("r", width/2)
        .style("fill-opacity","0")
        .transition()
        .duration(Math.ceil((Math.random()*3+2)*1000))
        .attr("r", "100")
        .on("end",function(){
            moveCircle(circle);
        });
}

for(var i = 0; i < 5; i++){
    for(var j = 0; j < 2; j++){
        var circle = svg.append("circle")
            .attr("cx",160*(i+0.5))
            .attr("cy",200*(j+1))
            .attr("r","100")
            .style("fill-opacity","0")
            .style("stroke","red")
            .style("stroke-width","3");
        moveCircle(circle);
    }
}
</script>
</body>
</html>

```

U kodu napravimo 10 kružnica u 2 reda po 5. Nakon toga moramo napraviti funkciju za beskonačnu petlju moveCircle() koja ima na kraju svog koda .on("end", function()) dio kojim možemo ponovno pozvati istu funkciju nad istim objektom. Nasumičnost smo postigli koristeći Math.random() u duration() dijelu koda. Animacija počne sinkrono ali nakon nekog vremena sve kružnice postanu asinkrone od jedna druge.

Zadatak 3.



Kod:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Z3-3</title>
  <script src="https://d3js.org/d3.v5.min.js"></script>
</head>
<body>
  <script>
    svg = d3.select("body")
      .append("svg")
      .attr("width", 800)
```

```

        .attr("height", 800)
        .style("background-color", "lightblue");

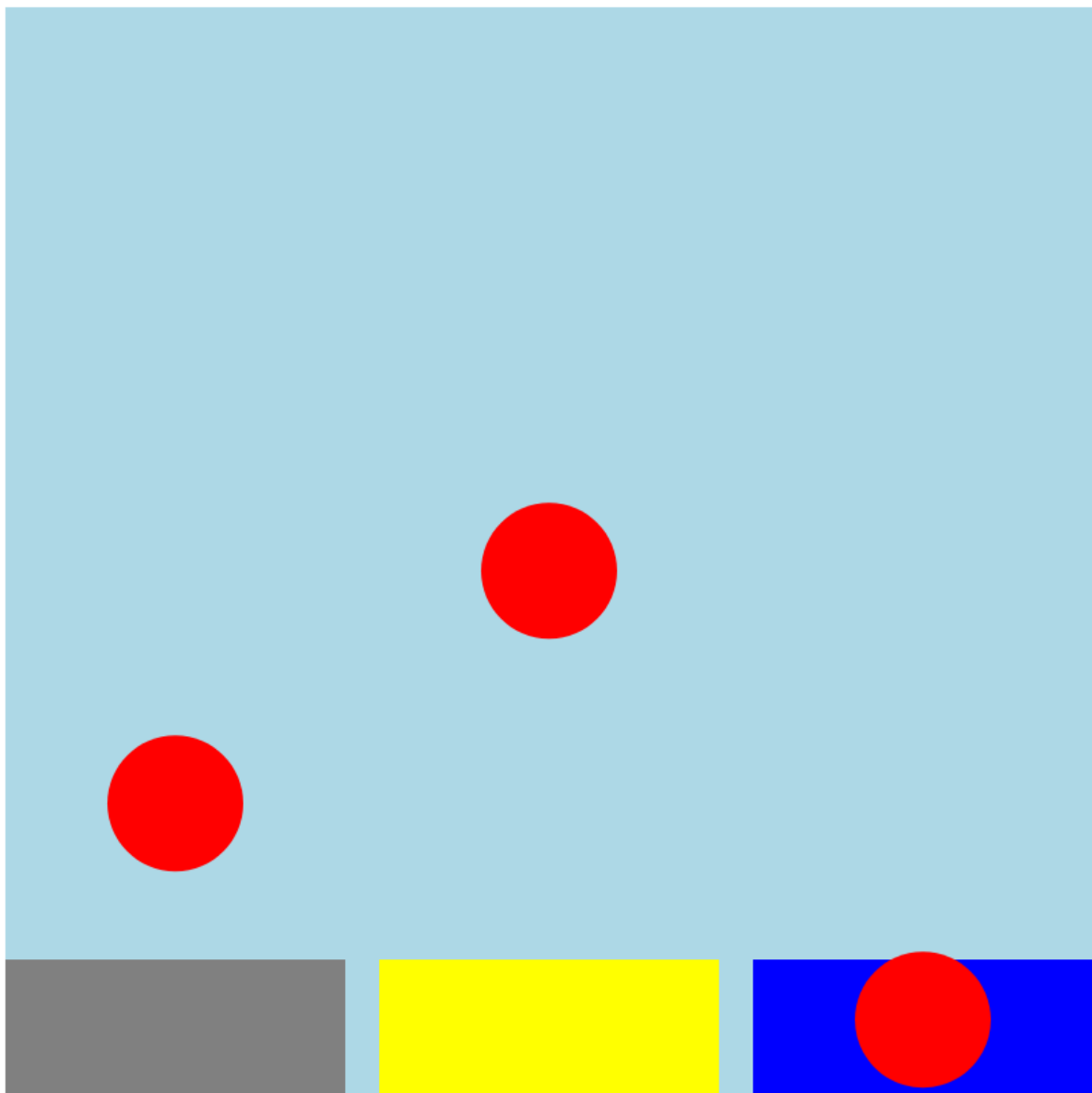
    rect = svg.append("rect")
        .attr("x", 0)
        .attr("y", 0)
        .attr("width", 200)
        .attr("height", 200)
        .style("fill", "green");

    rect.transition()
        .duration(1000)
        .attr("x", svg.attr("width") - rect.attr("width"))
        .transition()
        .duration(1000)
        .attr("y", svg.attr("height") - rect.attr("height"));
    </script>
</body>
</html>

```

Kod je dosta jednostavan, napravimo rect element te damo mu transition() koji ga prvo preko jedne sekunde pomakne iz početne (0,0) pozicije u skroz gornji desni kut svg elementa, pa onda opet preko jedne sekunde ga pomakne vertikalno prema doljnjem desnom kutu svg elementa.

Zadatak 4.



Kod:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Z3-4</title>
  <script src="https://d3js.org/d3.v5.min.js"></script>
</head>
<body>
  <script>
    svg = d3.select("body")
      .append("svg")
      .attr("width", 800)
```



```

        .attr("height", 800)
        .style("background-color", "lightblue");

    rect_solid = svg.append("rect")
        .attr("width", 250)
        .attr("height", 100)
        .attr("x", 0)
        .attr("y", svg.attr("height") - 100)
        .style("fill", "gray");

    rect_elastic = svg.append("rect")
        .attr("width", 250)
        .attr("height", 100)
        .attr("x", 275)
        .attr("y", svg.attr("height") - 100)
        .style("fill", "yellow");

    rect_liquid = svg.append("rect")
        .attr("width", 250)
        .attr("height", 100)
        .attr("x", 550)
        .attr("y", svg.attr("height") - 100)
        .style("fill", "blue");

    circle_solid = svg.append("circle")
        .attr("r", 50)
        .attr("cx", parseInt(rect_solid.attr("x")) + parseInt(rect_solid
.attr("width")/2))
        .attr("cy", 60)
        .style("fill", "red");

    circle_elastic = svg.append("circle")
        .attr("r", 50)
        .attr("cx", parseInt(rect_elastic.attr("x")) + parseInt(rect_ela
stic.attr("width")/2))
        .attr("cy", 60)
        .style("fill", "red");

    circle_liquid = svg.append("circle")
        .attr("r", 50)
        .attr("cx", parseInt(rect_liquid.attr("x")) + parseInt(rect_liqu
id.attr("width")/2))
        .attr("cy", 60)
        .style("fill", "red");

    function bounce(h) {
        if (!arguments.length) h = 0.25;
        var b0 = 1 - h,

```

```

        b1 = b0 * (1 - b0) + b0,
        b2 = b0 * (1 - b1) + b1,
        x0 = 2 * Math.sqrt(h),
        x1 = x0 * Math.sqrt(h),
        x2 = x1 * Math.sqrt(h),
        t0 = 1 / (1 + x0 + x1 + x2),
        t1 = t0 + t0 * x0,
        t2 = t1 + t0 * x1,
        m0 = t0 + t0 * x0 / 2,
        m1 = t1 + t0 * x1 / 2,
        m2 = t2 + t0 * x2 / 2,
        a = 1 / (t0 * t0);
        return function(t) {
            return t >= 1 ? 1
                : t < t0 ? a * t * t
                : t < t1 ? a * (t -= m0) * t + b0
                : t < t2 ? a * (t -= m1) * t + b1
                : a * (t -= m2) * t + b2;
        };
    }

    circle_solid.transition()
        .ease(bounce(0.15))
        .duration(3120)
        .attr("cy", parseInt(svg.attr("height")) - parseInt(rect_solid.attr("height")) - parseInt(circle_solid.attr("r")));

    circle_elastic.transition()
        .ease(bounce(0.4))
        .duration(5000)
        .attr("cy", parseInt(svg.attr("height")) - parseInt(rect_elastic.attr("height")) - parseInt(circle_elastic.attr("r")));

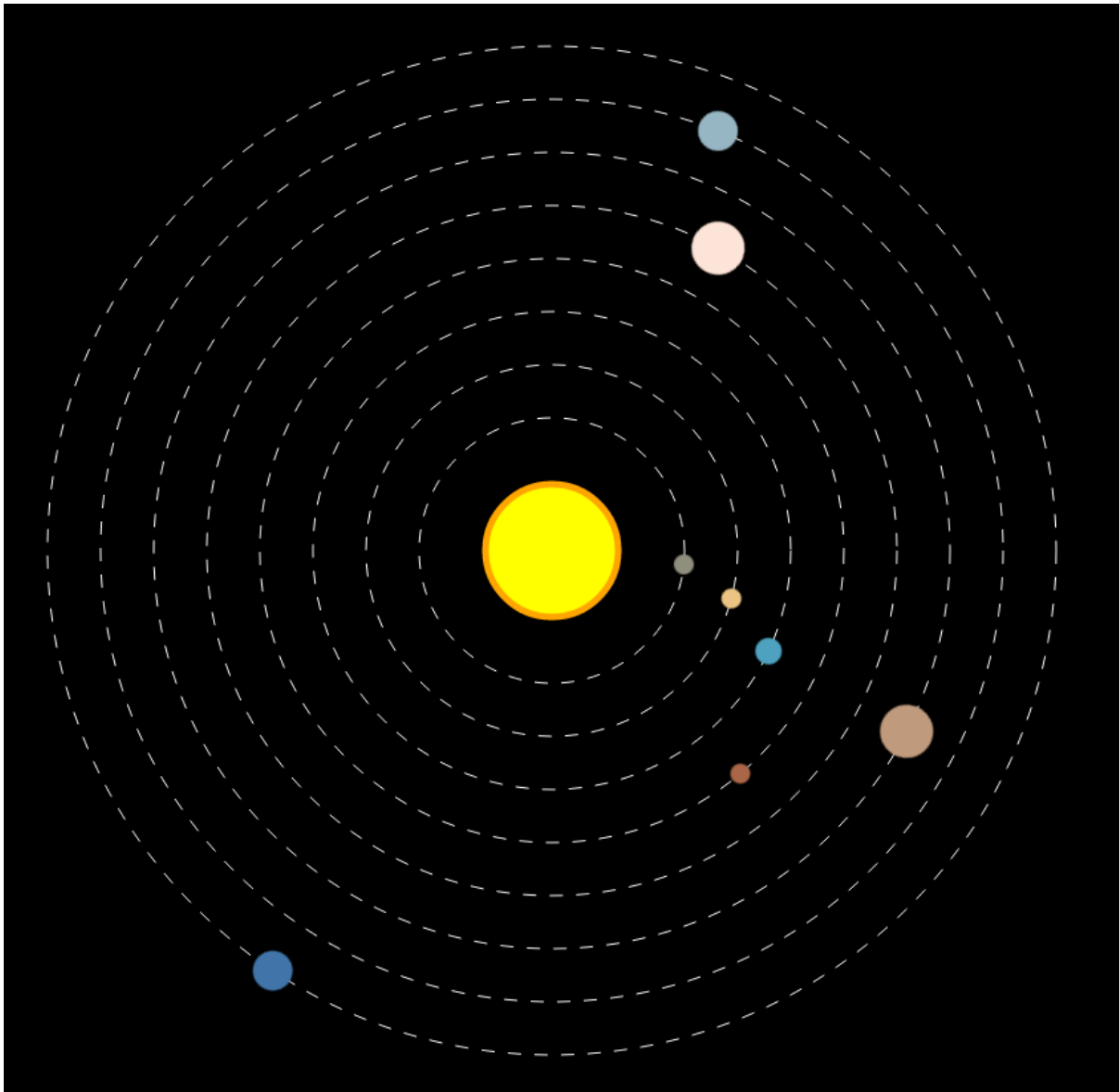
    circle_liquid.transition()
        .ease(d3.easeElasticOut)
        .duration(28000)
        .attr("cy", parseInt(svg.attr("height")) - parseInt(rect_liquid.attr("height")));
    </script>
</body>
</html>

```

Na početku kreiramo tri pravokutnika kao materijale pa jednu pripadajuću kružnicu svakom pravokutniku. Cilj je donekle simulirati materijal (krut, elastičan i tekući) te zato se koristi ease() funkcija kod transition() funkcije. U ease() funkciji možemo navesti neku metodu koja će opisati pokret kružnice da što bliže opiše pravo simulirani pokret. Za krutu tvar u ease() smo koristili prilagođenu bounce() funkciju kojom smo prigušili (smanjili faktor u funkciji) skakanje kružnice. Za elastičnu tvar u ease() smo koristili istu tu bounce() funkciju ali smanjili smo prigušenje (povećali

faktor u funkciji) te time postigli puno više skakanja kružnice nego kod krute tvari. Za tekućinu nismo mogli bounce koristiti pa smo koristili ugrađenu d3 funkciju `d3.easeElasticOut` da bi donekle simulirali skakanje kružnice kroz tekućinu. Kod svih funkcija smo `duration()` morali prilagoditi tako da se sve tri kružnice na početku prividno jednako padaju.

Zadatak 5.



Kod:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Z3-5</title>
  <script src="https://d3js.org/d3.v5.min.js"></script>
</head>
<body>
  <script>
    var globalSpeed = 0.1;
    var globalSize = 0.5;
    var data = [
      {"size":15, "speed":0.24, "color":"#908f7e", "rotation":0},
```

```

        {"size":15, "speed":0.6, "color":"#ebc483", "rotation":0},
        {"size":20, "speed":1, "color":"#4fa2bf", "rotation":0},
        {"size":15, "speed":2, "color":"#aa6646", "rotation":0},
        {"size":40, "speed":12, "color":"#fde4d8", "rotation":0},
        {"size":40, "speed":30, "color":"#bf9a7c", "rotation":0},
        {"size":30, "speed":84, "color":"#97b6c4", "rotation":0},
        {"size":30, "speed":164, "color":"#4175a9", "rotation":0},
    ]
    svg = d3.select("body")
        .append("svg")
        .attr("width", 800)
        .attr("height", 800)
        .style("background-color", "black")
        .style("margin", "20px");

    sun = svg.append("circle")
        .attr("cx", parseInt(svg.attr("width"))/2)
        .attr("cy", parseInt(svg.attr("height"))/2)
        .attr("r", 50)
        .style("fill", "yellow")
        .style("stroke", "orange")
        .style("stroke-width", "5");

    for(var i = 0; i<8 ; i++){
        svg.append("circle")
            .attr("cx", parseInt(svg.attr("width"))/2)
            .attr("cy", parseInt(svg.attr("height"))/2)
            .attr("r", 100+i*40)
            .style("fill-opacity", "0")
            .style("stroke", "white")
            .style("stroke-dasharray", "10, 10");
    }

    function rotatePlanet(planet, i){
        data[i].rotation += data[i].speed*globalSpeed;
        planet.transition()
            .duration(30)
            .attr("transform", "rotate("+ data[i].rotation +", 400, 400)
    ")
            .on("end", function(){
                rotatePlanet(planet, i);
            });
    }

    for(var i = 0; i < 8; i++){
        planet = svg.append("circle")
            .attr("cx", parseInt(svg.attr("width"))/2 + 100 + i*40)
            .attr("cy", parseInt(svg.attr("height"))/2)

```

```
        .attr("r", data[i].size * globalSize)
        .style("fill", data[i].color);
    rotatePlanet(planet, i);
}
d3.select("body")
    .style("background-color", "black");
</script>
</body>
</html>
```

Kreiramo sunce u centru, te u petlji kreiramo kružnice kao putanje i pripadajuće planete. Svaki planet ima svoja svojstva: veličina, brzina, boja i trenutna pozicija u rotaciji. U funkciji `rotatePlanet()` povlačimo trenutnu rotaciju planeta, uvećavamo za zadanu brzinu te usporimo nekom globalnom varijablom inače nam se simulacija prebrzo odvija. Onda preko `transition()` zadamo novu vrijednost koristeći `attr()` setter, te ponovno kao u drugom zadatku preko `on("end", function(){})` napravimo beskonačnu petlju. Na ovaj način smo simulirali kretanje sunčevog sustava, jedino što globalna brzina nije točna (zemlja ne traje 1 cijeli dan nego par minuta), ali relativna brzina među planetima je točna.