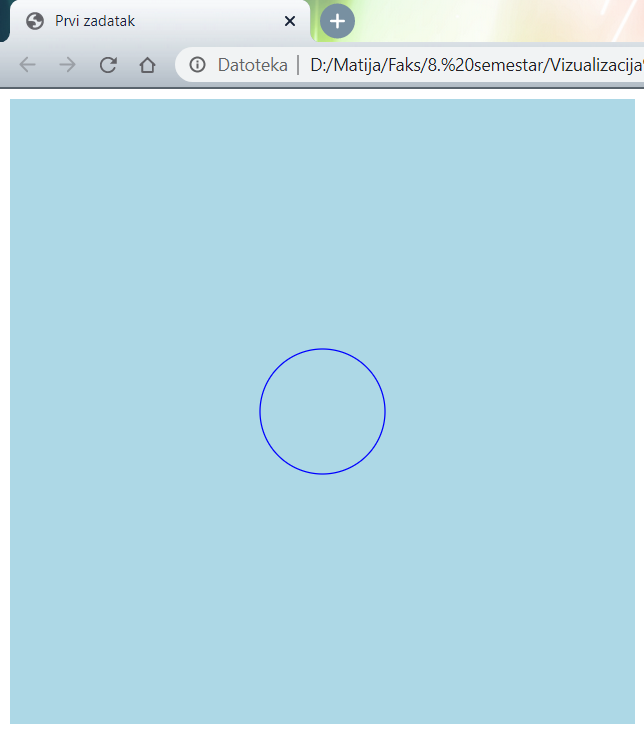
**Vizualizacija podataka - Laboratorijska vježba 3**

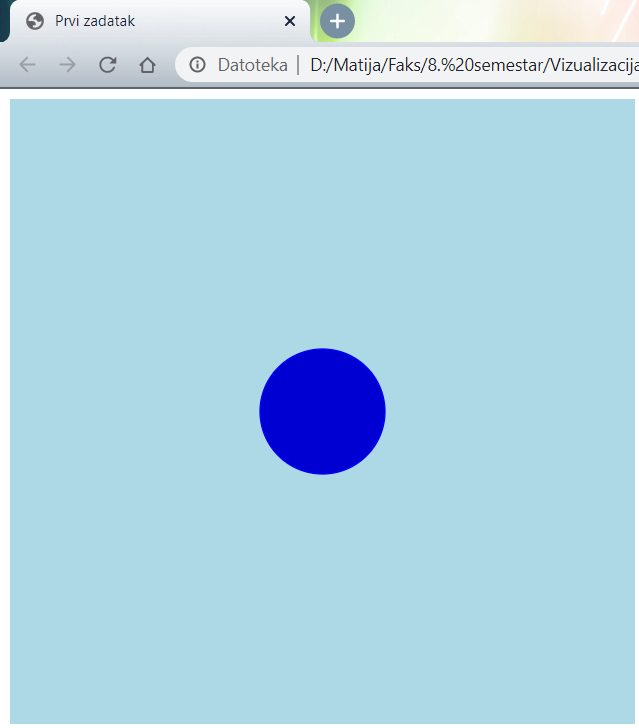
**Animacija i pridruživanje podataka**

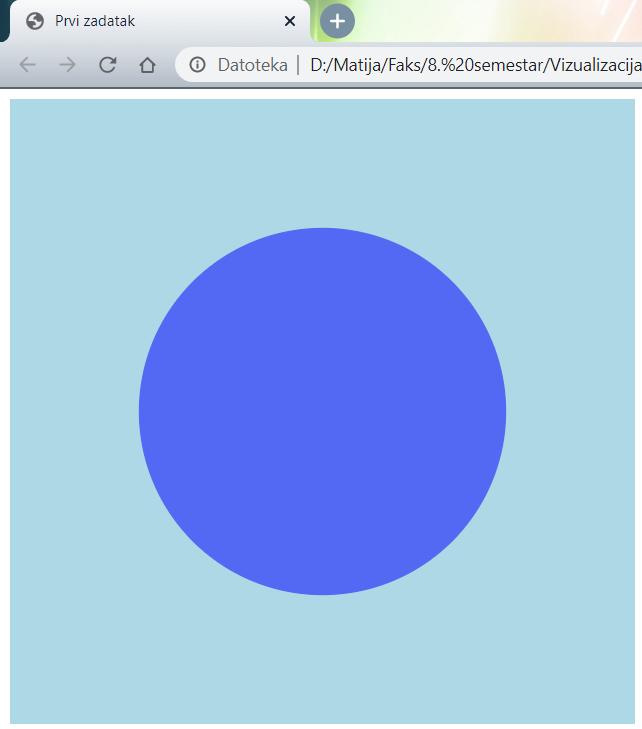
1. **Napisati kod koji će unutar SVG elementa umetnuti kružnicu. Kružnica treba biti bez ispune (engl. *fill*) s obrubom (engl. *stroke*) u boji. Nakon 3 s potrebno je pokrenuti tranziciju koja će dodati ispunu tijekom perioda od 5 s. Nakon završetka ispunjavanja kružnice potrebno je pokrenuti novu tranziciju koja će povećati radijus do vrijednosti *widthSVG* / 2, a pri tome istovremeno smanjivati parametar *opacity* do vrijednosti 0.**

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <script src="https://d3js.org/d3.v3.min.js"></script>  <title>Prvi zadatak</title>  </head>  <body>  <script>  var width = 500;  var height = 500;    var svg = d3.select("body")  .append("svg")  .attr("width", width)  .attr("height", height)  .style("background-color", "lightblue");    var circle = svg.append("circle")  .attr("cx", width / 2)  .attr("cy", height / 2)  .attr("r", 50)  .style("fill", "none")  .style("stroke", "blue");    svg.select("circle")  .transition()  .delay(3000)  .duration(5000)  .style("fill", "blue")  .style("stroke", "blue");    svg.select("circle")  .transition()  .delay(8000)  .duration(5000)  .attr("r", width / 2)  .style("opacity", 0);  </script>  </body>  </html> |

Izlistanje koda 1. Sadržaj Z3-1.html datoteke

  
Slika 1. Početno stanje prvog zadatka

   
Slika 2. Stanje tijekom promjene boje

  
Slika 3. Stanje tijekom povećanja i promjene prozirnosti

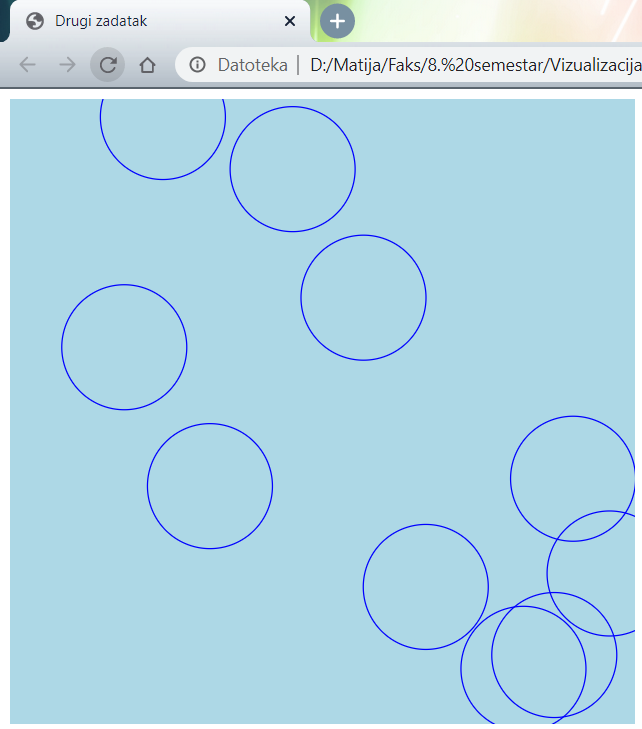
**Komentar:**

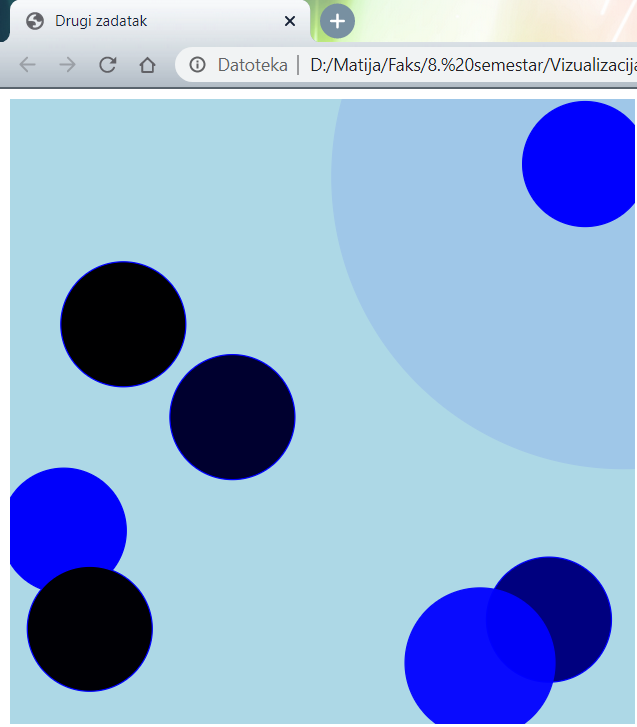
U ovome zadatku je nacrtan krug kojemu se nakon određenog vremena mijenja boja, veličina i prozirnosti. Sve funkcionalnosti su odrađene pomoću D3.js biblioteke. Rješenje zadataka je prikazano na 1., 2. i 3. slici.

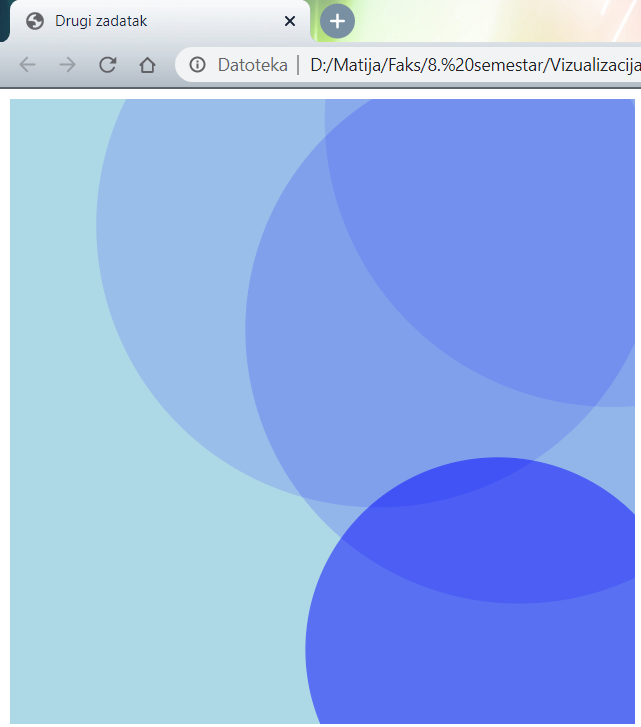
1. **Izmijeniti kod prethodnog zadatka na način da se omogući autonomno dodavanje 10 kružnica koje će se ponašati slično kao i u prethodnom zadatku, ali sa slučajno odabranom vrijednošću trajanja svake od tranzicija. Nakon isteka posljednje tranzicije postupak je potrebno ponoviti s novim vrijednostima vremena i položaja te početnim vrijednostima radijusa.**

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <script src="https://d3js.org/d3.v3.min.js"></script>  <title>Drugi zadatak</title>  </head>  <body>  <script>  var width = 500;  var height = 500;  var counter = 0;    var svg = d3.select("body")  .append("svg")  .attr("width", width)  .attr("height", height)  .style("background-color", "lightblue");  function update() {  var data = d3.range(10).map(function() { return { x: Math.random()\*500, y: Math.random()\*500 }; });  var transition = d3.range(10).map(function() { return { delay: Math.random()\*3000, duration: Math.random()\*3000 }; });  var circle = svg.selectAll("circle")  .data(data)  .enter()  .append("circle")  .attr("cx", function(d) { return d.x; })  .attr("cy", function(d) { return d.y; })  .attr("r", 50)  .style("fill", "none")  .style("stroke", "blue");  var trans = svg.selectAll("circle")  .data(transition)  .transition()  .each("start", function(){ counter++; })  .delay(function (d){ return d.delay; })  .duration(function (d){ return d.duration; })  .style("fill", "blue");  svg.selectAll("circle")  .data(transition)  .transition()  .delay(function (d) { return d.delay+d.duration; })  .duration(function (d) { return Math.ceil(Math.random()\*3000); })  .attr("r", width / 2)  .style("opacity", 0)  .each("end", function() {  d3.select(this).remove();  counter--;  if(counter == 0){  update();  }  });  }  update();  </script>  </body>  </html> |

Izlistanje koda 2. Sadržaj Z3-2.html datoteke

  
Slika 4. Početno stanje drugog zadatka

  
Slika 5. Stanje tijekom mijenjaja boja, veličine i prozirnosti

   
Slika 6. Stanje tijekom mijenjaja boja, veličine i prozirnosti

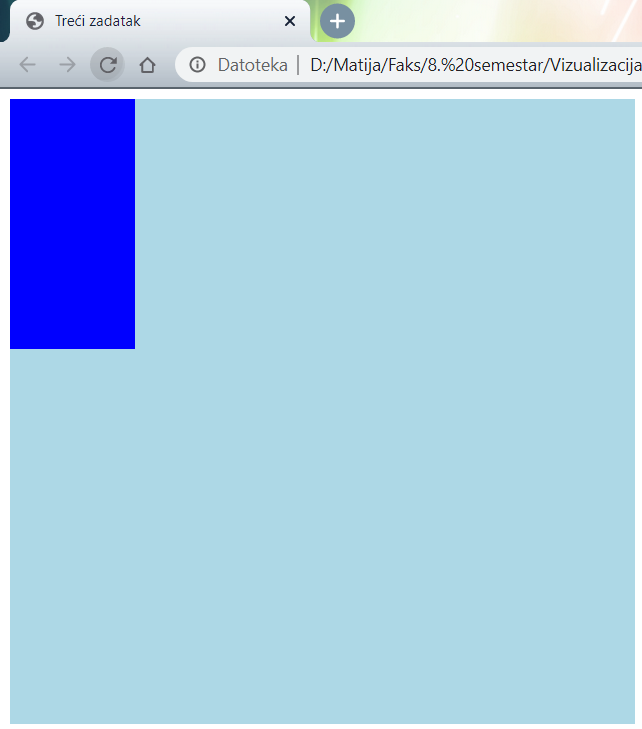
**Komentar:**

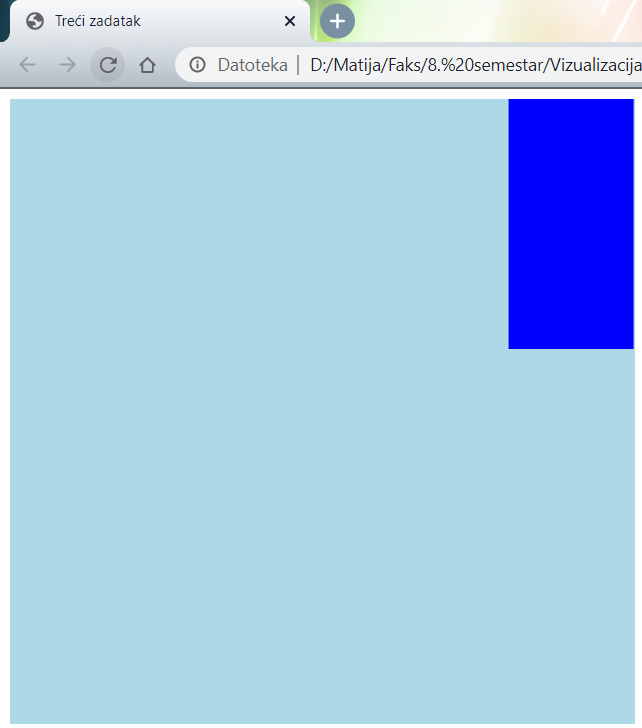
Za razliku od prethodnog zadatka, vrijeme potrebno za promjenu boje, povećanja i prozirnoti dobiveno je nasumičnim odabirom. Maksimalno vrijeme je 3 s. Nakon isteka posljednje tranzicije, postupak se ponavlja i tako beskonačno puta. Rješenja su prikazana na 4., 5. i 6. slici.

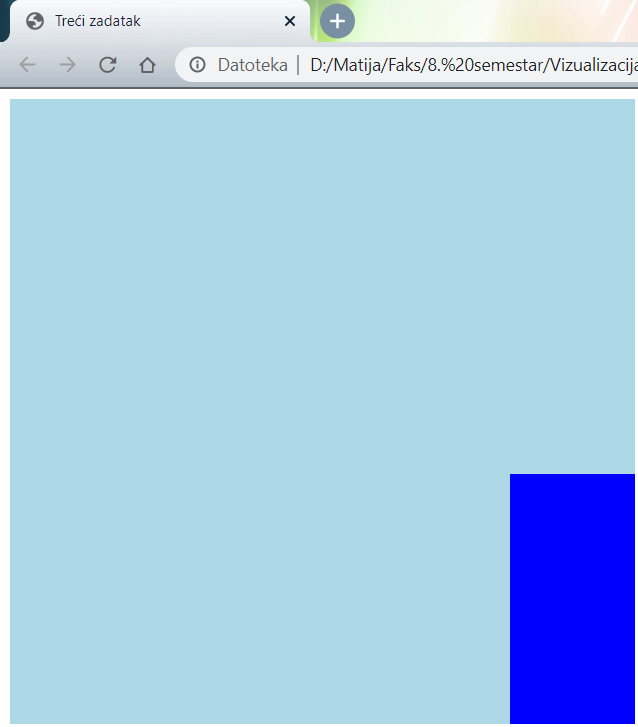
1. **Napisati kod koji će pomoću biblioteke D3 primjenom SVG transformacija translatirati umetnuti pravokutnik iz položaja (0, 0) u položaj (*widthSVG* – *widthrect*, *heightSVG* – *heightrect*), ali na način da se pravokutnik translatira prvo u smjeru x-osi, a potom u smjeru y-osi. Trajanje svake translacije treba biti 1 s.**

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <script src="https://d3js.org/d3.v3.min.js"></script>  <title>Treći zadatak</title>  </head>  <body>  <script>  var width = 500;  var height = 500;  var rectWidth = 100;  var rectHeight = 200;    var svg = d3.select("body")  .append("svg")  .attr("width", width)  .attr("height", height)  .style("background-color", "lightblue");  var rect = svg.append("rect")  .attr("x", 0)  .attr("y", 0)  .attr("width", rectWidth)  .attr("height", rectHeight)  .style("fill", "blue");  svg.select("rect")  .transition()  .delay(250)  .duration(1000)  .attr("transform", function(d) { return `translate(${width - rectWidth}, 0)`; } );  svg.select("rect")  .transition()  .delay(1250)  .duration(1000)  .attr("transform", function(d) { return `translate(${width - rectWidth}, ${height - rectHeight})`; } );  </script>  </body>  </html> |

Izlistanje koda 3. Sadržaj Z3-3.html datoteke

  
Slika 7. Početno stanje trećeg zadatka

  
Slika 8. Stanje nakon translatiranja po x-osi

  
Slika 9. Stanje nakon translatiranja po y-osi

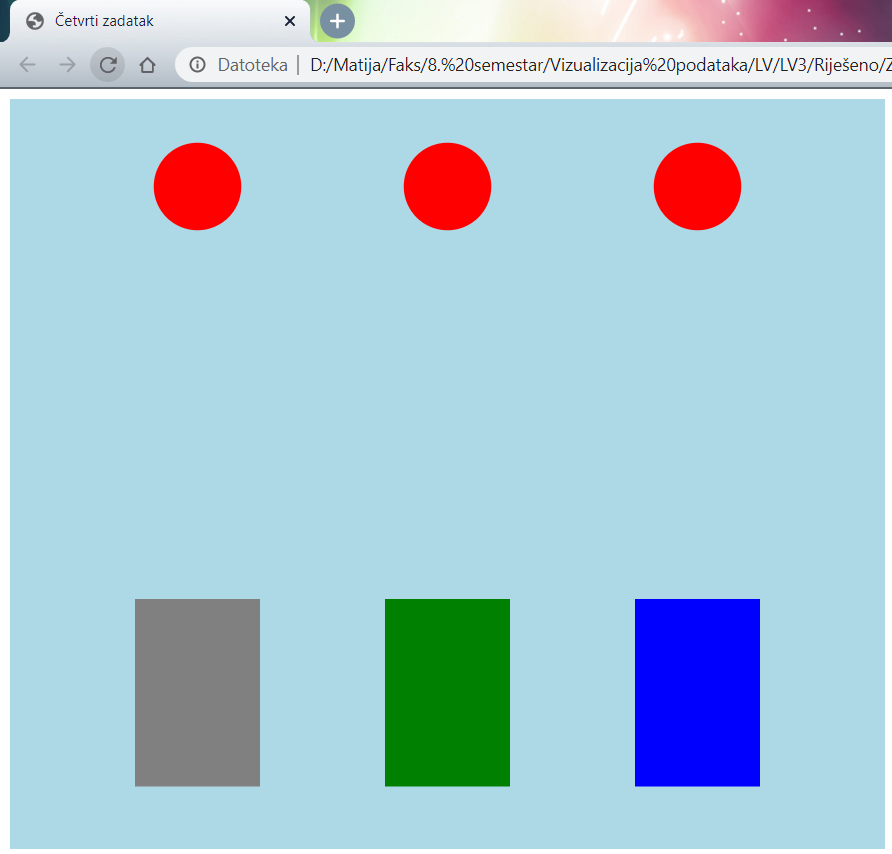
**Komentar:**

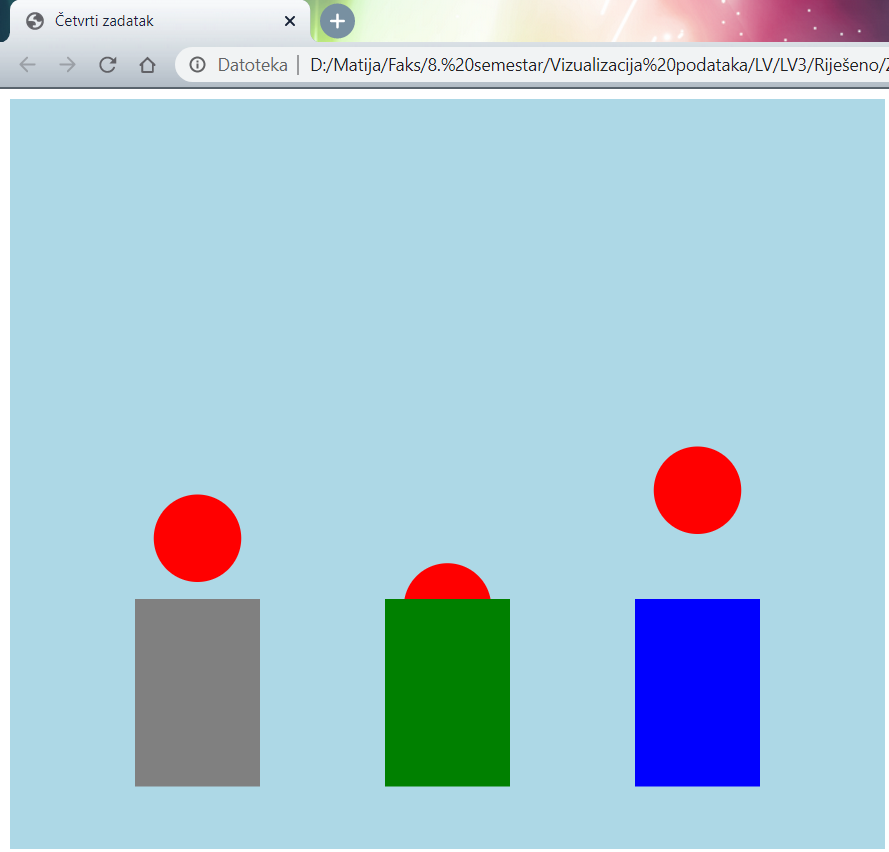
U zadatku je omogućena translacija pravokutnika po x-osi te nakon toga po y-osi. Svaka translacija traje 1 s. Potrebno je obratiti pažnju na kašnjene (engl. *delay*) druge translacije (po y-osi) kako bi ista krenula tek nakon izvršavanja translacije po x-osi. Rješenja su prikazana na 7., 8. i 9. slici.

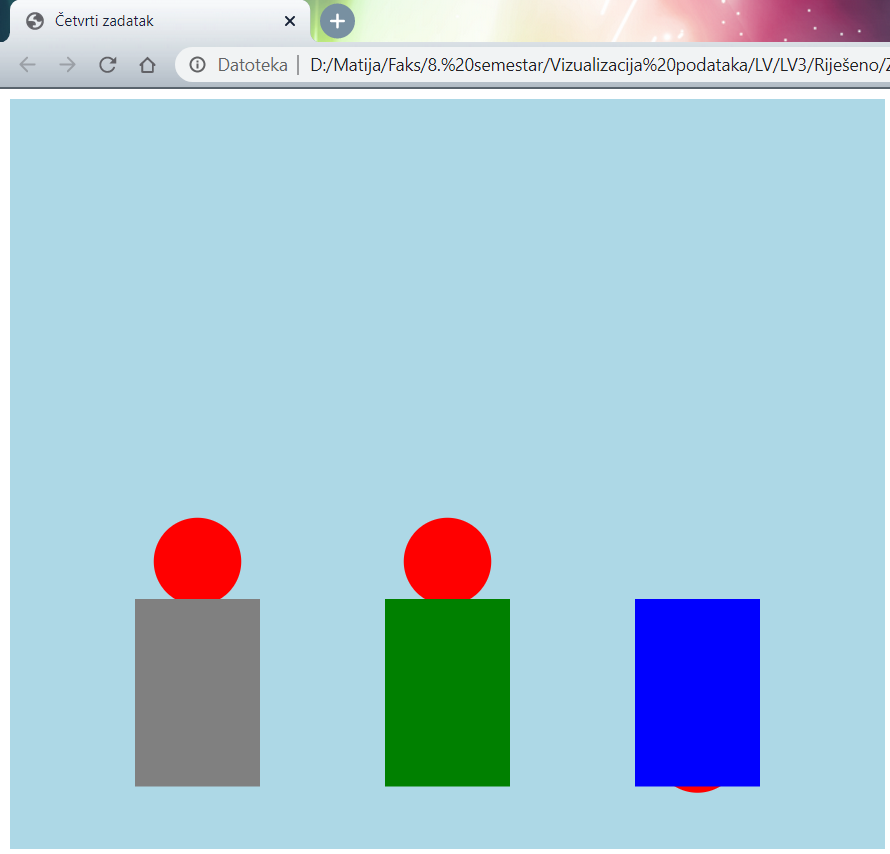
1. **U SVG umetnuti tri kružnice koje će biti pri vrhu slike. Pri dnu, a ispod svake od kružnica, umetnuti tri pravokutnika koji će predstavljati različite materijale i to redom krutu tvar, elastičnu tvar i tekućinu. Simulirati slobodan pad kugli i njihovu reakciju prilikom sudara s pravokutnicima.**

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <script src="https://d3js.org/d3.v3.min.js"></script>  <title>Četvrti zadatak</title>  </head>  <body>  <script>  var width = 700;  var height = 600;  var rectWidth = 100;  var rectHeight = 150;  var radius = 35;  var circleY = 70;  var colors = ["grey", "green", "blue"];  var svg = d3.select("body")  .append("svg")  .attr("width", width)  .attr("height", height)  .style("background-color", "lightblue");  for (i = 0; i < 3; i++) {  var id = "circle" + (i + 1);  var cx = 150 + i \* 200;  makeCircle(id, cx)  var x = 100 + i \* 200;  makeRectangle(x, colors[i])  }  makeTransformation("#circle1", 250, 1000, "bounce", "translate(0, 300)");  makeTransformation("#circle2", 250, 6000, "elastic", "translate(0, 300)");  makeTransformation("#circle3", 250, 1000, "linear-in-out", "translate(0, 300)");  makeTransformation("#circle3", 1250, 1000, "linear-in-out", "translate(0, 450)");  function makeCircle(id, cx) {  svg.append("circle")  .attr("id", id)  .attr("cx", cx)  .attr("cy", circleY)  .attr("r", radius)  .style("fill", "red");  }  function makeRectangle(x, color) {  svg.append("rect")  .attr("x", x)  .attr("y", 400)  .attr("width", rectWidth)  .attr("height", rectHeight)  .style("fill", color);  }  function makeTransformation(id, delay, duration, ease, transform) {  svg.select(id)  .transition()  .delay(delay)  .duration(duration)  .ease(ease)  .attr("transform", transform);  }  </script>  </body>  </html> |

Izlistanje koda 4. Sadržaj Z3-4.html datoteke

  
Slika 10. Početno stanje četvrtog zadatka

  
Slika 11. Stanje tijekom pada kugli

  
Slika 12. Konačno stanje

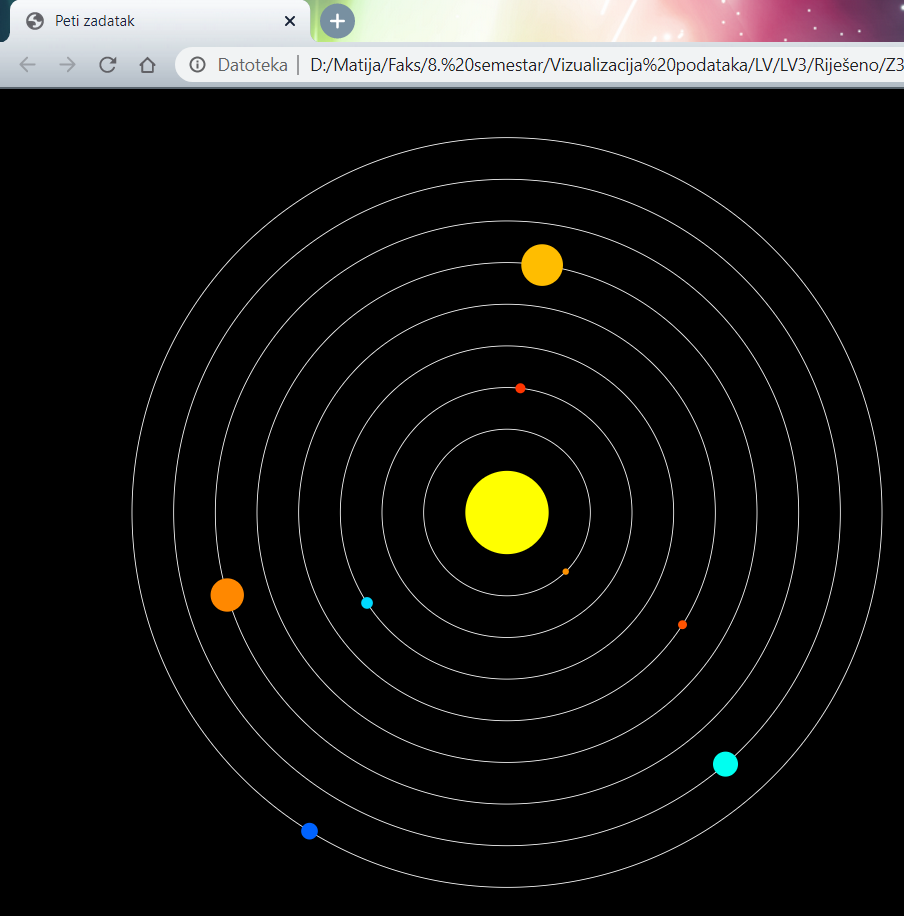
**Komentar:**

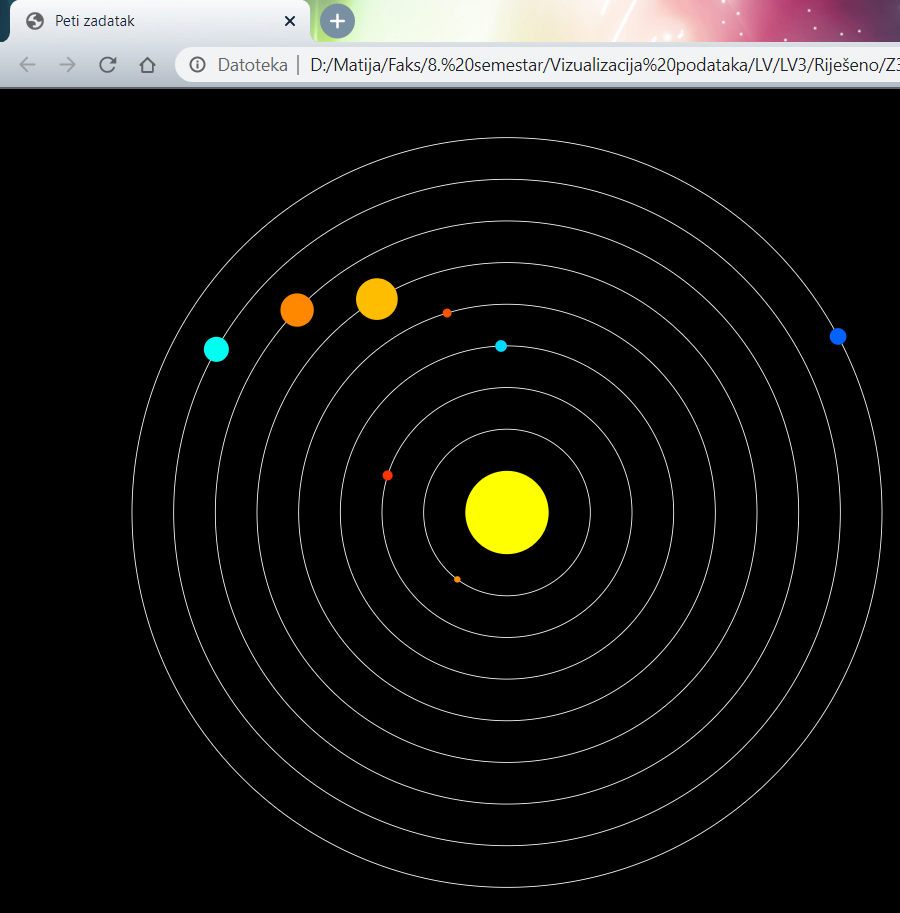
U zadatku je bilo potreno mijenjati parametar ease() funkcije. Svaka kugla pada na različitu površinu (pravokutnik) stoga je svaka imala drugačiji parametar(*bounce*, *elastic*, *linear-in-out*). Zbog različitog parametra, svaka kugla će različito reagirati u dodiru s površinom. Rješenja su prikazana na 10., 11. i 12. slici.

1. **Izraditi simulaciju sunčevog sustava s pripadajućim rotacijama planeta primjenom biblioteke D3.**

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <script src="https://d3js.org/d3.v3.min.js"></script>  <title>Peti zadatak</title>  <style>  body {  background-color: black;  }  </style>  </head>  <body>  <script>  var width = 1200;  var height = 1000;  var sunRadius = 50;  var planets = [{distance: 100, speed: 20, radius: 3.8, position: 100, color: "#FF9200"}, //mercury  {distance: 150, speed: 16, radius: 6, position: 200, color: "#FF3500"}, //venera  {distance: 200, speed: 12, radius: 7, position: 300, color: "#00D9FF"}, //earth  {distance: 250, speed: 10, radius: 5.3, position: 400, color: "#FF5300"}, //mars  {distance: 300, speed: 8, radius: 25, position: 500, color: "#FFBD00"}, //jupiter  {distance: 350, speed: 6, radius: 20, position: 600, color: "#FF8800"}, //saturn  {distance: 400, speed: 4, radius: 15, position: 700, color: "#00FFF0"}, //uran  {distance: 450, speed: 3, radius: 10, position: 800, color: "#0063FF"}]; //neptun  var svg = d3.select("body")  .append("svg")  .attr("width", width)  .attr("height", height)  svg.append("circle")  .attr("r", sunRadius)  .attr("cx", width / 2)  .attr("cy", height / 2)  .attr("fill", "yellow");  var group = svg.append("g")  .attr("transform", `translate(${width / 2}, ${height / 2})`);  group.selectAll("g.circle")  .data(planets)  .enter()  .append("g")  .each(function(d, i)  {  d3.select(this)  .append("circle")  .attr("class", "background")  .attr("r", d.distance)  .style("fill", "none")  .style("stroke", "white");  d3.select(this)  .append("circle")  .attr("r", d.radius)  .attr("cx", d.distance)  .attr("cy", 0)  .attr("class", "rotate\_circle")  .style("fill", d.color);  });  var time = Date.now();  d3.timer(function() {  var difference = Date.now() - time;  svg.selectAll(".rotate\_circle")  .attr("transform", function(d) {  return `rotate(-${d.position + difference \* d.speed / 300})`;  });  });  </script>  </body>  </html> |

Izlistanje koda 5. Sadržaj Z3-5.html datoteke

  
Slika 13. Prikaz petog zadatka

  
Slika 14. Prikaz petog zadatka

**Komentar:**

U realizaciji ovog zadatka korišteno je polje koje je sadržavalo bitne podatke za svaki planet (veličina, udaljenost, boja,...). Kružnice, koje predstavljaju putanju kretanja planeta, napravljene su kao obični *circle* element, samo bez ispune, ali s obojanim rubom. Na svaku kružnicu dodan je *circle* element koji predstavlja planet. Kako bi bilo omogućeno kretanje planeta, korišten je *timer* koji se periodično okida te se izvodi dio koda potreban za simulaciju kretanja planeta.