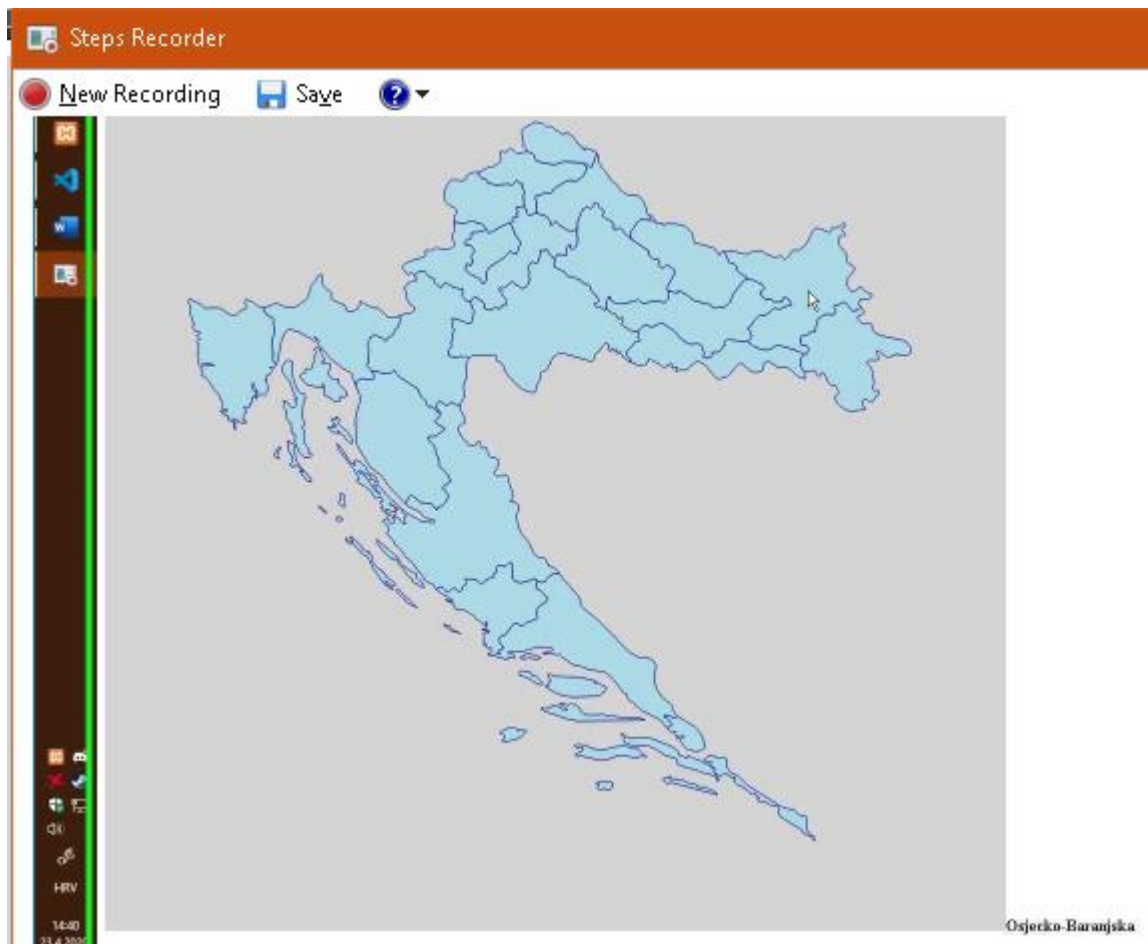


Laboratorijske vježbe iz predmeta Vizualizacija Podataka

Zadatak 1.



Kod:

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

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Z5-1</title>
  <script src="https://d3js.org/d3.v5.min.js"></script>
  <script src="https://d3js.org/topojson.v1.min.js"></script>
</head>

<body>
  <script>
    var width = 900;
    var height = 900;
    var projection = d3.geoAlbers().parallels([40, 50])
      .center([0.5, 44.5])
      .rotate([-16, 0])
      .scale(10000)
      .translate([width / 2, height / 2]);
```

```

var path = d3.geoPath().projection(projection);

var svg = d3.select("body")
  .append("svg")
  .attr("width", width)
  .attr("height", height)
  .style("background", "lightgray");

var tooltip = d3.select("body")
  .append("div")
  .style("opacity", 0)
  .style("width", 600)
  .style("display", "inline-block");

fetch("cro_regv3.json")
  .then(response => response.json())
  .then(json => {
    var data = topojson.feature(json, json.objects.layer1);
    var states = svg.selectAll("path.county")
      .data(data.features)
      .enter()
      .append("path")
      .attr("class", "county")
      .attr("id", function (d) { return d.id; })
      .attr("d", path)
      .style("fill", "lightblue")
      .style("stroke", "navy")
      .style("stroke-width", 1)
      .style("stroke-opacity", 1)
      .on("mouseover", function (d) {
        console.log(d.properties.name);
        var tooltip_tip = "<h4>" + d.properties.name + "</h4>";

        tooltip.html(tooltip_tip);

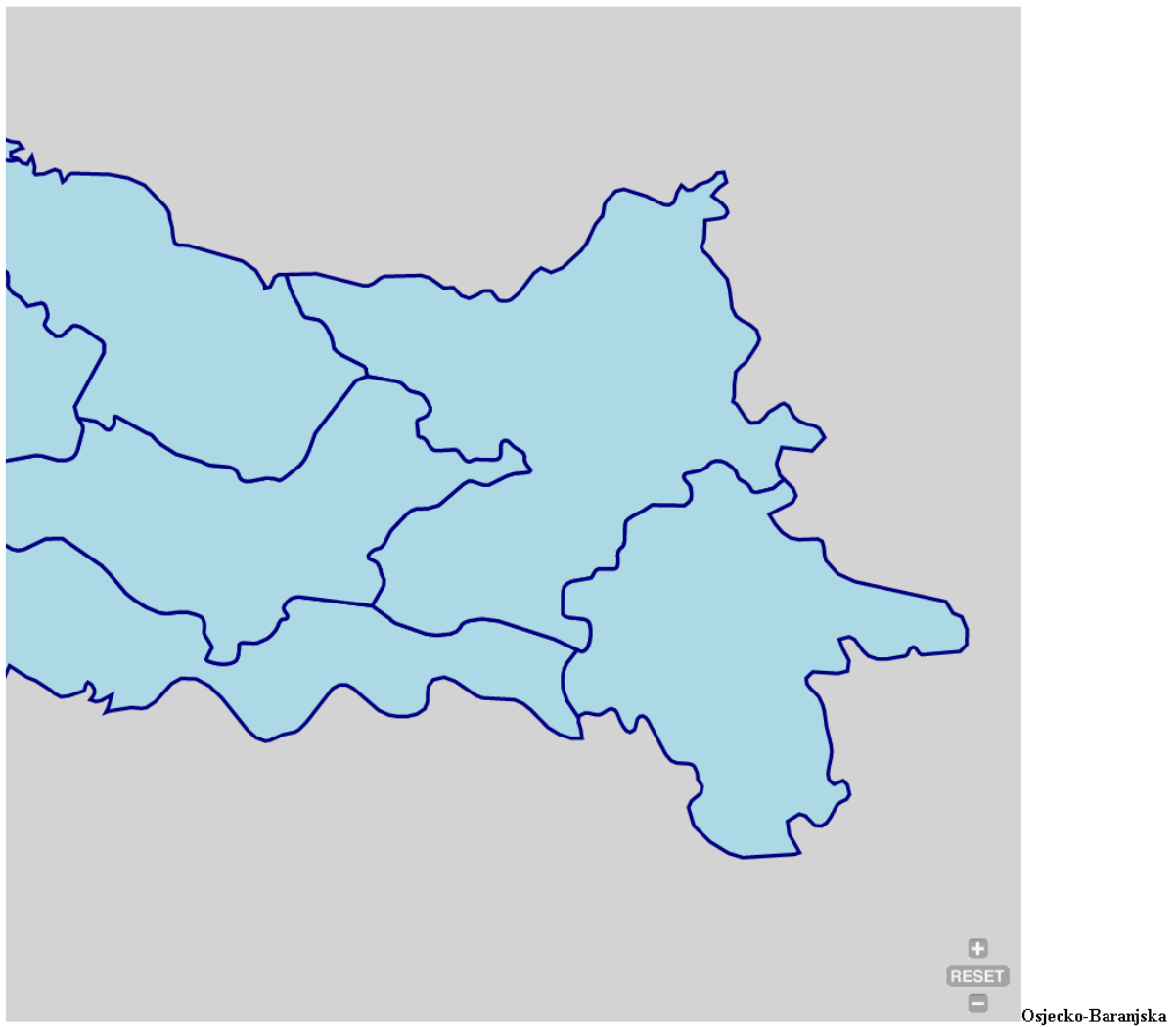
        tooltip.transition()
          .duration(100)
          .style("opacity", 1);
      })
      .on("mouseout", function (d) {
        tooltip.transition()
          .duration(1000)
          .style("opacity", 0);
      });
  });
});
</script>
</body>

```

```
</html>
```

Koristeći d3 biblioteku kreiramo potrebne osnovni svg element u kojeg ćemo stavljati ostale elemente. Za prikaz ovakvih podataka potrebna nam je projekcija koju možemo odabrati (postoji više različitih vrsta) preko d3 biblioteke. Zatim kreiramo generator krivina path koji prima odabranu projekciju koju smo prethodno naveli. U fetch bloku učitavamo podatke iz .json datoteke te podatke obrađujemo po predlošku. Dodan je tekstualni blok koji će imati ispis imena županije kad se prijeđe mišem preko. Svaka županija je dobila svoj on mouseover i mouseout slušatelj koji govori kako se mora tekstualni blok ponašati.

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>Z5-2</title>
  <script src="https://d3js.org/d3.v5.min.js"></script>
  <script src="https://d3js.org/topojson.v1.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/svg-pan-zoom@3.5.0/dist/svg-pan-zoom.min.js"></script>
</head>

<body>
  <script>
    var width = 900;
    var height = 900;
```

```

var projection = d3.geoAlbers().parallels([40, 50])
    .center([0.5, 44.5])
    .rotate([-16, 0])
    .scale(10000)
    .translate([width / 2, height / 2]);
var path = d3.geoPath().projection(projection);

var svg = d3.select("body")
    .append("svg")
    .attr("width", width)
    .attr("height", height)
    .attr("id", "svg_pan_zoom")
    .style("background", "lightgray");

var tooltip = d3.select("body")
    .append("div")
    .style("opacity", 0)
    .style("width", 600)
    .style("display", "inline-block");

fetch("cro_regv3.json")
    .then(response => response.json())
    .then(json => {
        var data = topojson.feature(json, json.objects.layer1);
        var states = svg.selectAll("path.county")
            .data(data.features)
            .enter()
            .append("path")
            .attr("class", "county")
            .attr("id", function (d) { return d.id; })
            .attr("d", path)
            .style("fill", "lightblue")
            .style("stroke", "navy")
            .style("stroke-width", 1)
            .style("stroke-opacity", 1)
            .on("mouseover", function (d) {
                console.log(d.properties.name);
                var tooltip_tip = "<h4>" + d.properties.name + "</h4>";

                tooltip.html(tooltip_tip);

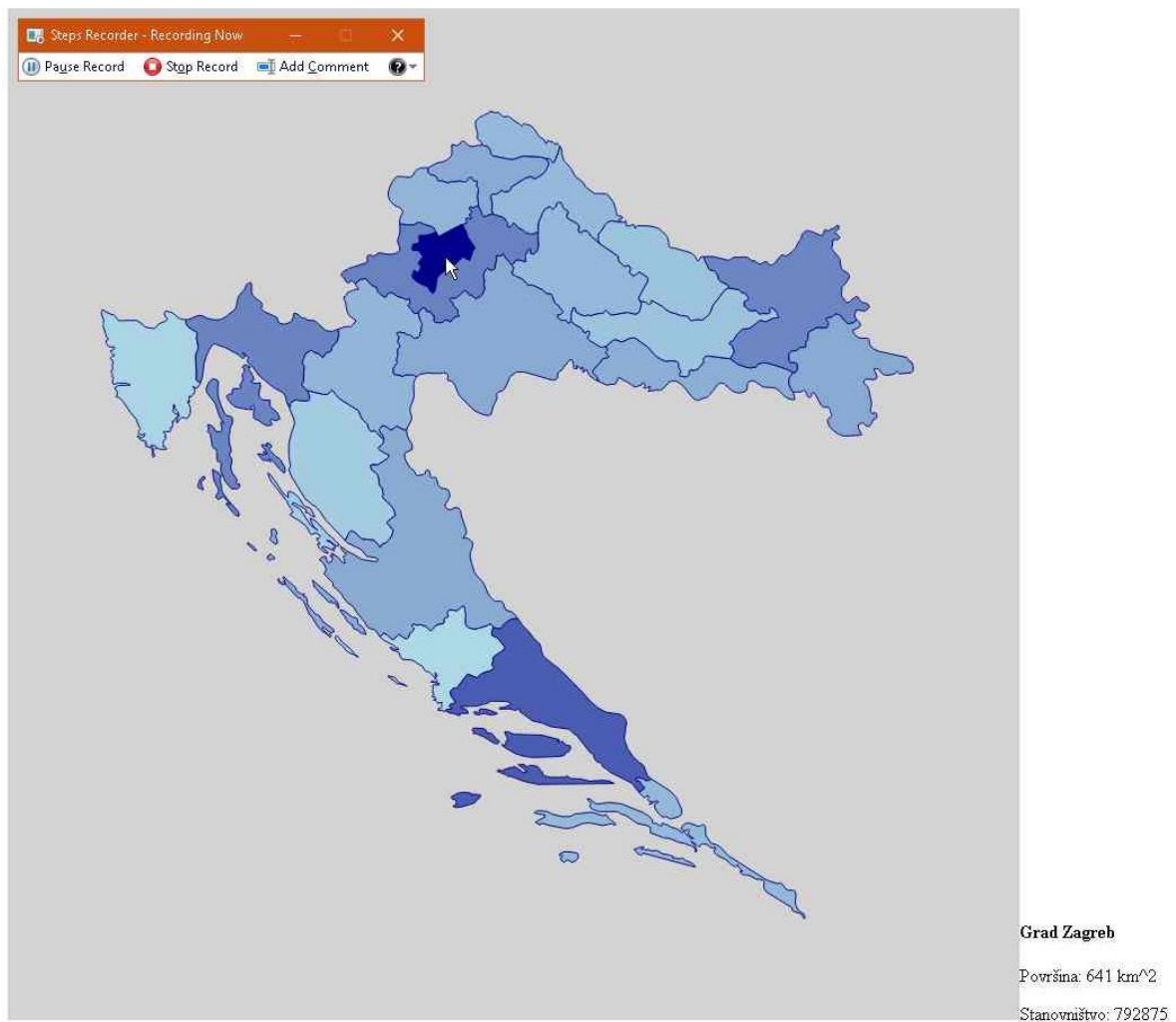
                tooltip.transition()
                    .duration(100)
                    .style("opacity", 1);
            })
            .on("mouseout", function (d) {
                tooltip.transition()
                    .duration(1000)

```

```
        .style("opacity", 0);
    });
    var zoomSVG = svgPanZoom('#svg_pan_zoom', {
        zoomEnabled: true,
        controlIconsEnabled: true,
        fit: true,
        center: true
    });
    });
</script>
</body>
</html>
```

Zadatak je identičan prethodnom zadatku sa razlikom da se koristi dodatna biblioteka za lagano namještanje micanja, zumiranja i manipuliranja prikaza svg prozora `svg-pan-zoom`. Na kraju `fetch` bloka se nalazi kod za inicijalizaciju djelovanja biblioteke nad svg prozorom. Moguće je mišem micati prikaz okolo, ili kotačićem na mišu zumirati vani i unutra. Duplim klikom se brzo uzumira. Dodatno su omogućene kontrole od kojih je jedna tipka reset za ponovno postavljanje prikaza na prvobitni položaj.

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>Z5-3</title>
  <script src="https://d3js.org/d3.v5.min.js"></script>
  <script src="https://d3js.org/topojson.v1.min.js"></script>
</head>

<body>
  <script>
    var hr_stats = [
      {
        "Županija": "Dubrovačko-neretvanska",
        "Sjedište": "Dubrovnik",
```



```
    "Površina (km2)": 1783,  
    "Stanovništvo": 122783  
  },  
  {  
    "Županija": "Šibensko-kninska",  
    "Sjedište": "Šibenik",  
    "Površina (km2)": 2939,  
    "Stanovništvo": 10932  
  },  
  {  
    "Županija": "Splitsko-dalmatinska",  
    "Sjedište": "Split",  
    "Površina (km2)": 4534,  
    "Stanovništvo": 455242  
  },  
  {  
    "Županija": "Zadarska",  
    "Sjedište": "Zadar",  
    "Površina (km2)": 3642,  
    "Stanovništvo": 170398  
  },  
  {  
    "Županija": "Krapinsko-zagorska",  
    "Sjedište": "Krapina",  
    "Površina (km2)": 1224,  
    "Stanovništvo": 133064  
  },  
  {  
    "Županija": "Karlovačka",  
    "Sjedište": "Karlovac",  
    "Površina (km2)": 3622,  
    "Stanovništvo": 128749  
  },  
  {  
    "Županija": "Ličko-senjska",  
    "Sjedište": "Gospić",  
    "Površina (km2)": 535,  
    "Stanovništvo": 51022  
  },  
  {  
    "Županija": "Primorsko-goranska",  
    "Sjedište": "Rijeka",  
    "Površina (km2)": 3582,  
    "Stanovništvo": 310195  
  },  
  {  
    "Županija": "Sisačko-moslavačka",  
    "Sjedište": "Sisak",
```

```
    "Površina (km2)": 4463,  
    "Stanovništvo": 172977  
  },  
  {  
    "Županija": "Zagrebačka",  
    "Sjedište": "Zagreb",  
    "Površina (km2)": 3078,  
    "Stanovništvo": 317642  
  },  
  {  
    "Županija": "Grad Zagreb",  
    "Sjedište": "Zagreb",  
    "Površina (km2)": 641,  
    "Stanovništvo": 792875  
  },  
  {  
    "Županija": "Istarska",  
    "Sjedište": "Pazin",  
    "Površina (km2)": 282,  
    "Stanovništvo": 20844  
  },  
  {  
    "Županija": "Brodsko-posavska",  
    "Sjedište": "Slavonski Brod",  
    "Površina (km2)": 2043,  
    "Stanovništvo": 158559  
  },  
  {  
    "Županija": "Osječko-baranjska",  
    "Sjedište": "Osijek",  
    "Površina (km2)": 4152,  
    "Stanovništvo": 304899  
  },  
  {  
    "Županija": "Požeško-slavonska",  
    "Sjedište": "Požega",  
    "Površina (km2)": 1845,  
    "Stanovništvo": 78031  
  },  
  {  
    "Županija": "Vukovarsko-srijemska",  
    "Sjedište": "Vukovar",  
    "Površina (km2)": 2448,  
    "Stanovništvo": 180117  
  },  
  {  
    "Županija": "Virovitičko-podravska",  
    "Sjedište": "Virovitica",
```

```

        "Površina (km2)": 2068,
        "Stanovništvo": 84586
    },
    {
        "Županija": "Bjelovarsko-bilogorska",
        "Sjedište": "Bjelovar",
        "Površina (km2)": 2652,
        "Stanovništvo": 119743
    },
    {
        "Županija": "Koprivničko-križevačka",
        "Sjedište": "Koprivnica",
        "Površina (km2)": 1746,
        "Stanovništvo": 115582
    },
    {
        "Županija": "Međimurska",
        "Sjedište": "Čakovec",
        "Površina (km2)": 730,
        "Stanovništvo": 114414
    },
    {
        "Županija": "Varaždinska",
        "Sjedište": "Varaždin",
        "Površina (km2)": 1261,
        "Stanovništvo": 176046
    }
];
var width = 900;
var height = 900;
var projection = d3.geoAlbers().parallels([40, 50])
    .center([0.5, 44.5])
    .rotate([-16, 0])
    .scale(10000)
    .translate([width / 2, height / 2]);
var path = d3.geoPath().projection(projection);

var min, max;
min = hr_stats[0]["Stanovništvo"];
max = hr_stats[0]["Stanovništvo"];
hr_stats.forEach(element => {
    if (element["Stanovništvo"] > max)
        max = element["Stanovništvo"];
    if (element["Stanovništvo"] < min)
        min = element["Stanovništvo"];
});
var color = d3.scaleLinear().domain([min, max]).range(["lightblue",
"darkblue"]);

```

```

var svg = d3.select("body")
  .append("svg")
  .attr("width", width)
  .attr("height", height)
  .style("background", "lightgray");

var tooltip = d3.select("body")
  .append("div")
  .style("opacity", 0)
  .style("width", 600)
  .style("display", "inline-block");

fetch("cro_regv3.json")
  .then(response => response.json())
  .then(json => {
    var data = topojson.feature(json, json.objects.layer1);
    var states = svg.selectAll("path.county")
      .data(data.features)
      .enter()
      .append("path")
      .attr("class", "county")
      .attr("id", function (d) { return d.id; })
      .attr("d", path)
      .style("fill", function (d, i) {
        return color(hr_stats[i]["Stanovnistvo"]);
      })
      .style("stroke", "navy")
      .style("stroke-width", 1)
      .style("stroke-opacity", 1)
      .on("mouseover", function (d, i) {
        console.log(d.properties.name);
        var tooltip_tip = "<h4>" + d.properties.name + "</h4>"
        + " km^2</p>" +
        "<p>Površina: " + hr_stats[i]["Površina (km2)"]
        + " km^2</p>" +
        "<p>Stanovništvo: " + hr_stats[i]["Stanovnistvo"]
        + "</p>";
        tooltip.html(tooltip_tip);

        tooltip.transition()
          .duration(100)
          .style("opacity", 1);
      })
      .on("mouseout", function (d, i) {
        tooltip.transition()
          .duration(1000)
          .style("opacity", 0);
      })
  })

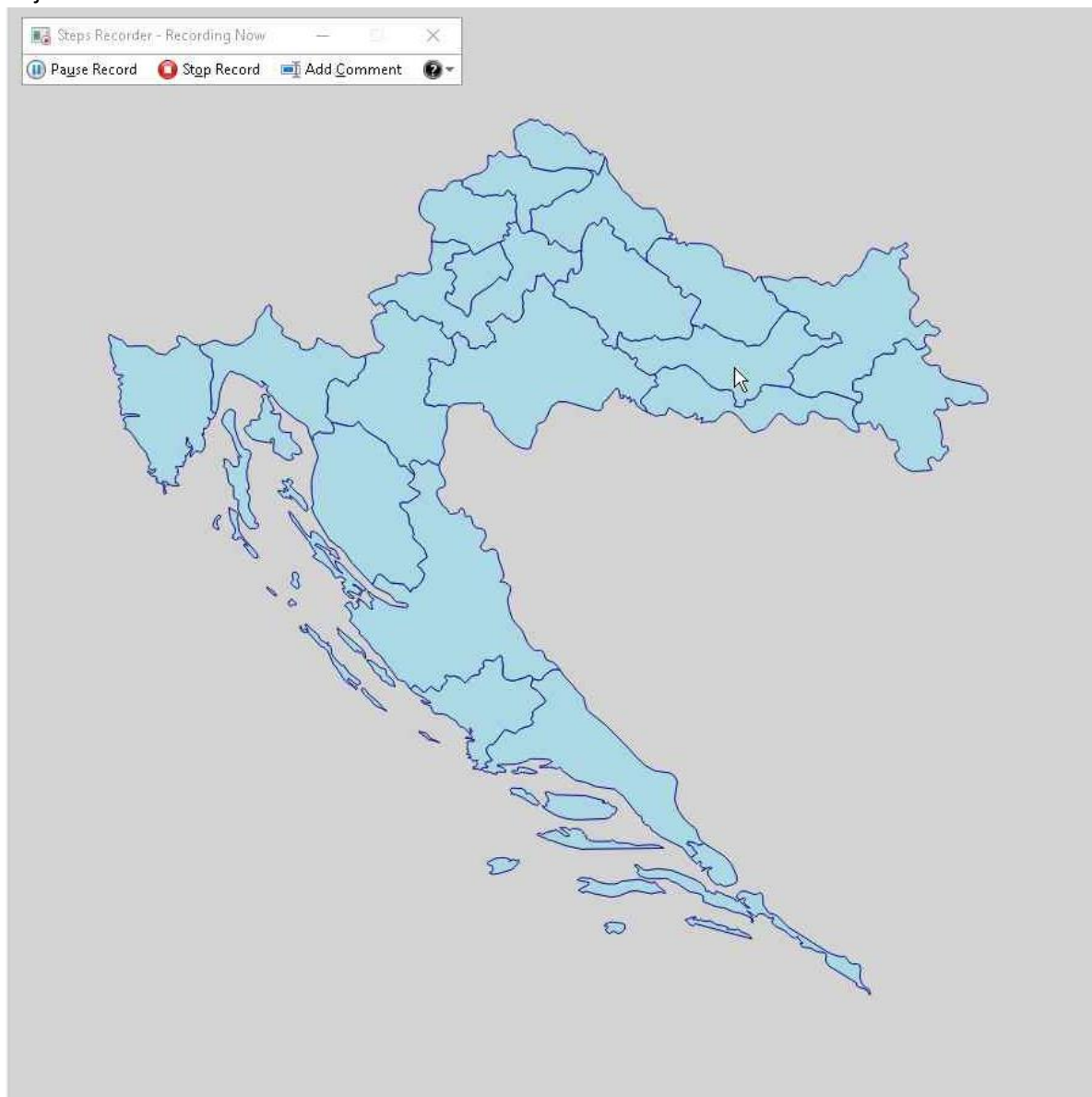
```

```
        });  
    });  
</script>  
</body>  
</html>
```

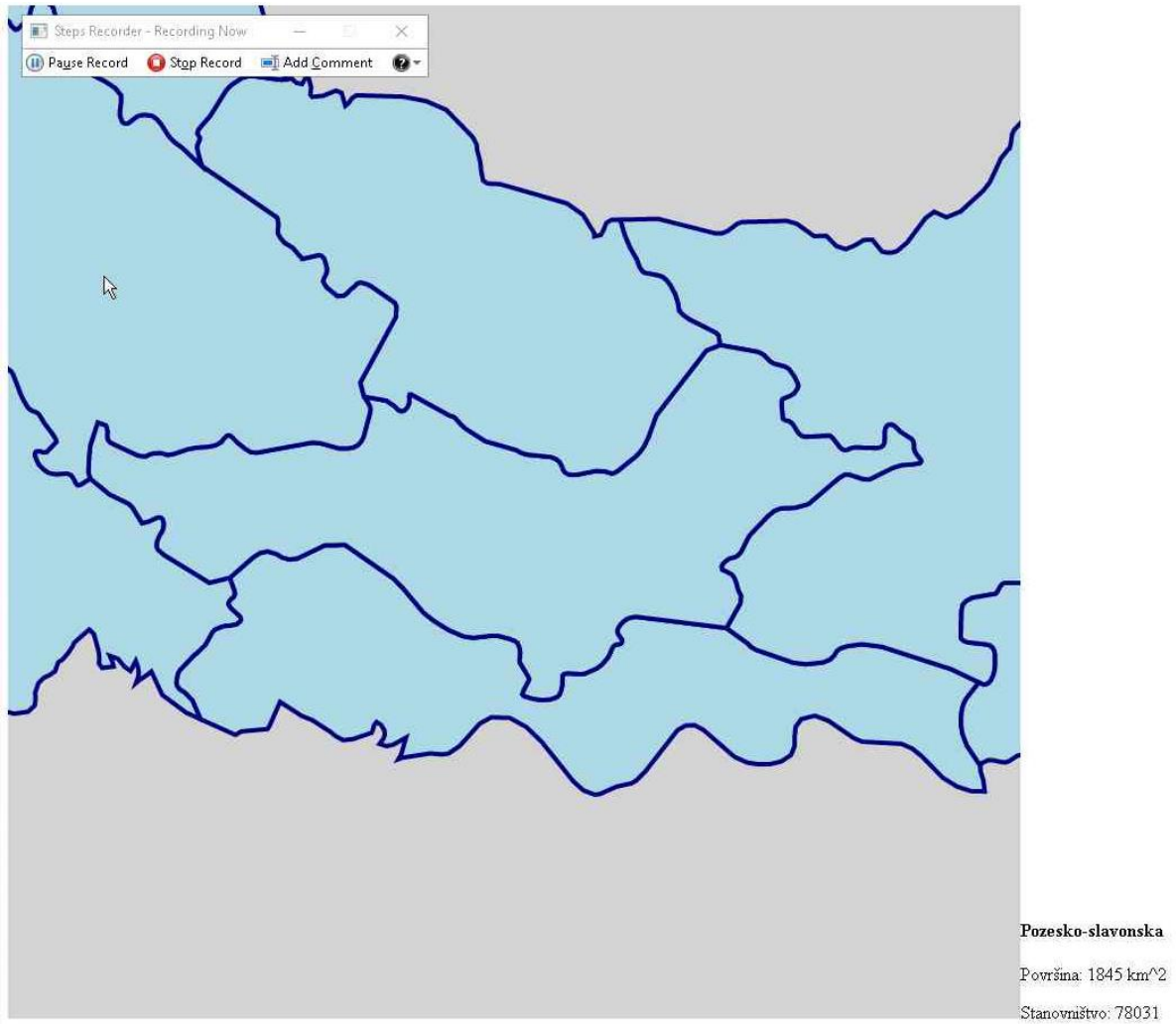
Zadatak je sličan prvom zadatku s razlikom da su dodane neke dodatne komponente koje smo obradili na prošlim laboratorijskim vježbama kao što su d3 color skale. Izrađena je skala za od svijetlo plave do tamno plave u ovisnosti o najmanjem stanovništvu i najvećem stanovništvu Hrvatskih županija. Nakon toga je za boju ispune odabrana boja koristeći navedenu skalu boja. Za konkretne dodatne informacije o županijama dodano je više informacija u prikazu teksta koji se vidi kad se mišem pređe preko županija.

Zadatak 4.

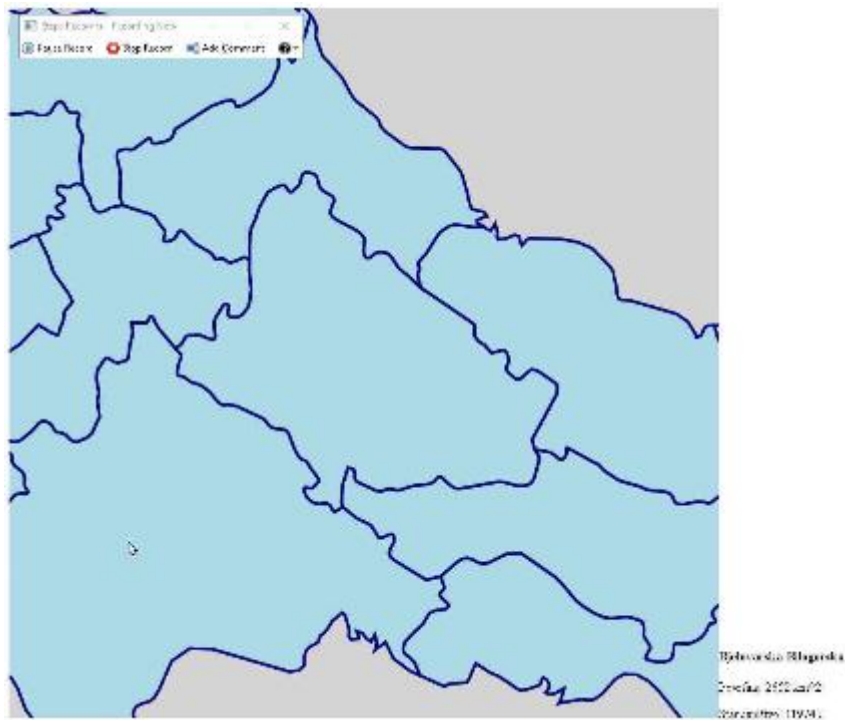
Prije klika:



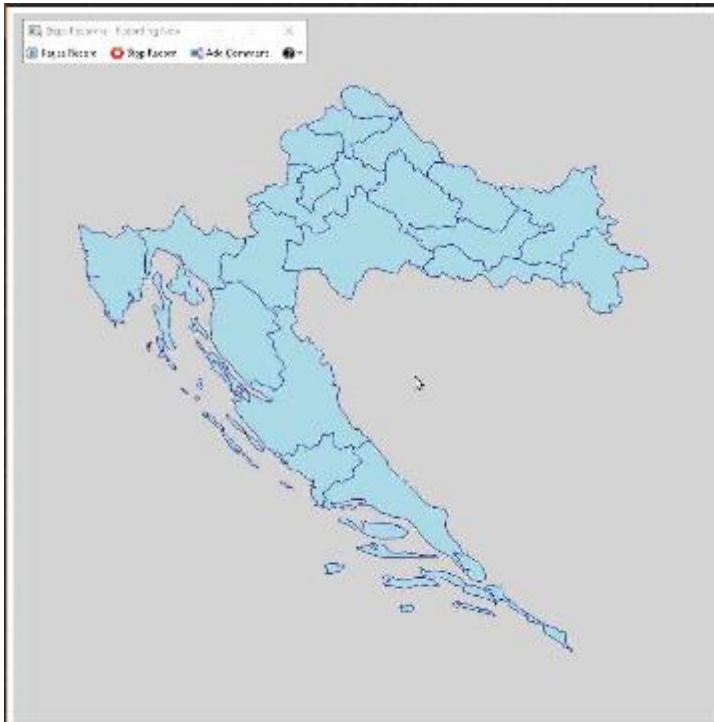
Nakon klika:



Nakon klika na drugu županiju:



Nakon 10 sekundi nakon zadnjeg klika:



Kod:

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



```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Z5-4</title>
<script src="https://d3js.org/d3.v5.min.js"></script>
<script src="https://d3js.org/topojson.v1.min.js"></script>
</head>
```

```
<body>
```

```
  <script>
```

```
    var hr_stats = [
```

```
      isto kao prošli zadatak -
```

```
    ];
```

```
    var width = 900;
```

```
    var height = 900;
```

```
    var centered;
```

```
    var projection = d3.geoAlbers().parallels([40, 50])
```

```
      .center([0.5, 44.5])
```

```
      .rotate([-16, 0])
```

```
      .scale(10000)
```

```
      .translate([width / 2, height / 2]);
```

```
    var path = d3.geoPath().projection(projection);
```

```
    var svg = d3.select("body")
```

```
      .append("svg")
```

```
      .attr("width", width)
```

```
      .attr("height", height)
```

```
      .style("background", "lightgray");
```

```
    var tooltip = d3.select("body")
```

```
      .append("div")
```

```
      .style("opacity", 0)
```

```
      .style("width", 600)
```

```
      .style("display", "inline-block");
```

```
    var counties = svg.append("g");
```

```
    function onZoomEnd(d) {
```

```
      tooltip.transition()
```

```
        .duration(1000)
```

```
        .style("opacity", 0);
```

```
    }
```

```
    fetch("cro_regv3.json")
```

```
      .then(response => response.json())
```

```
      .then(json => {
```

```
        var data = topojson.feature(json, json.objects.layer1);
```

```
        var states = counties.selectAll("path.county")
```

```
          .data(data.features)
```

```

        .enter()
        .append("path")
        .attr("class", "county")
        .attr("id", function (d) { return d.id; })
        .attr("d", path)
        .style("fill", "lightblue")
        .style("stroke", "navy")
        .style("stroke-width", 1)
        .style("stroke-opacity", 1)
        .on("click", function (d, i) {
            console.log("Clicked on " + d.properties.name);
            var x, y, k;
            var centroid = path.centroid(d);
            x = centroid[0];
            y = centroid[1];
            k = 4;
            centered = d;

            var tooltip_tip = "<h4>" + d.properties.name + "</h4>"
>" +
                "<p>Površina: " + hr_stats[i]["Površina (km2)"]
+ " km^2</p>" +
                "<p>Stanovništvo: " + hr_stats[i]["Stanovništvo"
] + "</p>";

            tooltip.html(tooltip_tip);

            tooltip.transition()
                .duration(100)
                .style("opacity", 1);

            console.log("X:" + x);
            console.log("Y:" + y);
            counties.transition()
                .duration(1000)
                .attr("transform", "translate(" + width / 2 + ",
" + height / 2 + ")scale(" + k + ")translate(" + -x + "," + -y + ")")
                .transition()
                .duration(10000)
                .transition()
                .duration(1000)
                .attr("transform", "translate(0,0) scale(1)")
                .on("end", onZoomEnd);

        });

    });
</script>
</body>

```

```
</html>
```

Kod je sličan prvom zadatku sa razlikom da ima on click slušatelj na svakoj županiji. Prilikom klika se povuku informacije o županiji na prikazu kao njena centralna lokacija. Koristeći te podatke se prikaz pomakne i uzumira preko jedne sekunde te pokaže se i dodatni tekst. Nakon 10 sekundi se prikaz vrati na normalno i tekst se makne koristeći on end slušatelj na tranzicijama. Moguće je kliknuti na jednu županiju, te onda kliknuti na drugu županiju i prebacit će prikaz na drugu županiju. Pritome klik od prve županije neće utjecati na klik od druge županije, tj. nakon 10 sekundi od prvog klika neće se vratiti natrag na staro nego će se tek vratiti nakon 10 sekundi od drugog klika na drugu županiju. Moguće je da svaki element prati svoje tranzicije, pa kad se on click slušatelj opali da osvježi sve trenutne tranzicije i stare makne, te zbog toga ne može prijevremeno se vratiti natrag na običan prikaz.