

### 3. KV3 - Izrada prototipne vizualizacije podataka

[Razrada koncepta, definiranje funkcionalnosti i ponašanja - izrada prototipa.]

#### 3.1. Osnovne funkcionalnosti i ponašanja

[Navesti osnovne funkcionalnosti vizualizacije i njihovo ponašanje]

- Z-3.1.1. Identificirati ključne funkcionalnosti koje će biti potrebne za prikaz podataka.

Za prikaz podataka koristit će se karta svijeta i linijski dijagram.

- Z-3.1.2. Definirati osnovne vrste ponašanja.

Korisnicima će biti omogućen odabir zemlje te godina za koju se podaci prikazuju. Također, na linijskom grafu bit će prikazan trend rasta za cijeli vremenski period.

- Z-3.1.3. Odabrati elemente s kojima će korisnici moći vršiti interakciju i definirati interakcije između korisnika i vizualizacije s pripadajućim opisom.

Korisnička interakcija ostvaruje se klikom na zemlju za odabir te klizač za podešavanje godine.

#### 3.2. Napredne funkcionalnosti i ponašanja:

[Navesti napredne funkcionalnosti vizualizacije i njihovo ponašanje]

- Z-3.2.1. Identificirati napredne funkcionalnosti koje će biti potrebne za analizu podataka.

- Z-3.2.2. Definirati napredne vrste ponašanja

- Z-3.2.3. Definirati interakcije koje će omogućiti korisnicima dodatnu analizu podataka.

Pružene napredne funkcionalnosti uključuju kretanje po karti (pan, zoom) te ispisivanje više različitih podataka istovremeno.

### 3.3. Implementacija osnovnih funkcionalnosti

[Opisati i dokazima potkrijepiti proces implementacije osnovnih funkcionalnosti]

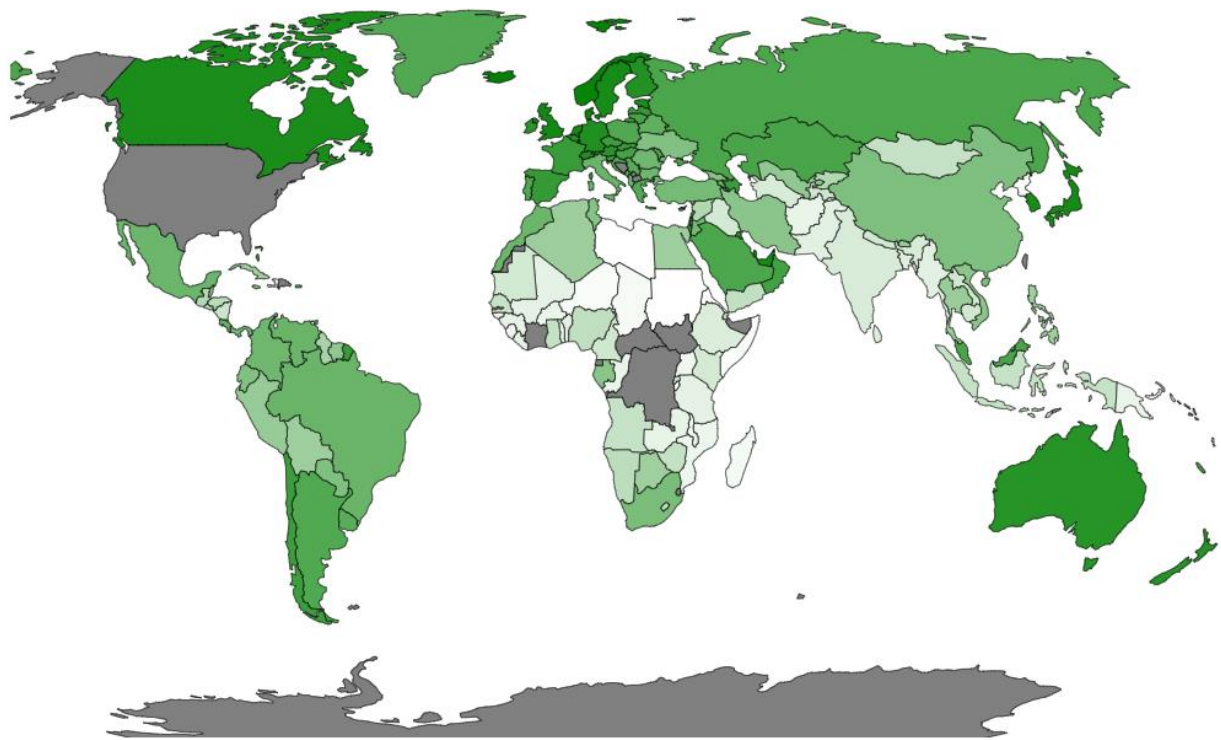
- Z-3.3.1. Izraditi kod koji omogućava prethodno definirane funkcionalnosti. Dokazati opisom primjera koda.

```
function updateMap() {
  const selectedYear = parseInt(yearInput.value) || 1980;
  const selectedVariable = document.getElementById("variable").value;

  d3.csv("assets/internet_percent.csv", (d) => {
    return {
      Entity: d.Entity,
      Code: d.Code,
      Year: +d.Year,
      InternetUsersPercentage: +d["Internet Users(%)"],
      UrbanPopulationPercentage: +d["Urban population(%)"]
    };
  }).then((data) => {
    const filteredData = data.filter((d) => d.Year === selectedYear);

    g.selectAll("path")
      .style("fill", (d) => {
        const countryData = filteredData.find((row) => row.Entity === d.properties.NAME);
        if (countryData) {
          const value = countryData[selectedVariable];
          return colorScale(value);
        } else {
          return "gray";
        }
      });
  });
}
```

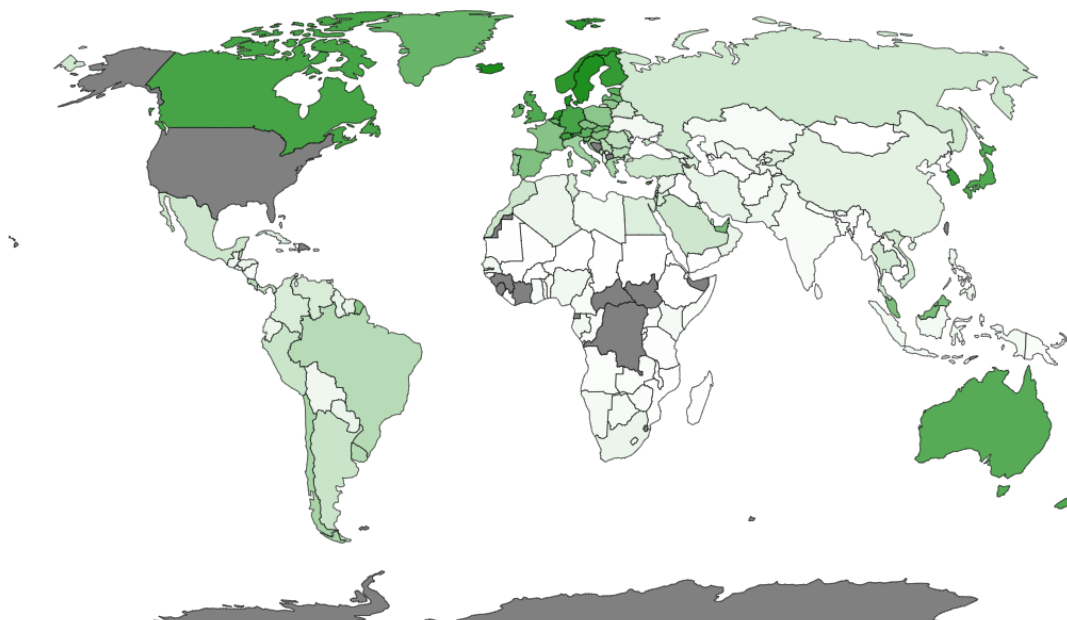
- Z-3.3.2. Testirati funkcionalnosti i osigurati da su ispravne, tj. da funkcioniraju na očekivani način. Potrebno potvrditi slikom.



### 3.4. Implementacija osnovnog ponašanja

[Opisati i dokazima potkrijepiti proces implementacije osnovnog ponašanja]

Z-3.4.1. Izraditi kod koji omogućava ranije definirano ponašanje. Dokazati opisom primjera koda.



Z-3.4.2. Testirati ponašanje i osigurati da je ispravno, tj. da funkcionira na očekivani način. Potrebno potvrditi slikom.

```
4. <div>
5.     <label for="year">Year:</label>
6.     <input type="range" id="year" min="1980" max="2020" step="1"
7.     value="2020">
8.     <span id="selectedYear">1980</span>
    </div>
```

Podaci se pravilno učitavaju iz skupa podataka. Potrebno je ostvariti potpunu funkciju klizača za odabir godine.