INF554 lab 7

Animated transitions

Dr. Luciano Nocera
Associate Director, Integrated Media Systems Center (IMSC)

USC Viterbi
School of Engineering
Integrated Media Systems Center

Catalina (new MacOS)

Reinstall xcode tools for git to work in VSCode by running in the terminal:

```
xcode-select --install
```

In your A7 repository create an ex1.html.

- Use data from your A1 for 10 countries and 1 year and load as JSON with d3:

```
data = [{key: "United States", value: 0.4},  //example data object after load
 {key: "Turkey", value: 0.6},
 {key: "France", value: 0.2},
 {key: "Italy", value: 0.1},
 {key: "Spain", value: 0.3},
 {key: "Algeria", value: 0.9},
 {key: "Mexico", value: 0.4},
 {key: "Brazil", value: 0.3},
 {key: "China", value: 0.1},
 {key: "Japan", value: 0.8}]
```

- Use d3 and `JSON.stringify` to print in the page:
    1. A copy of the array (use `array.slice`)
    2. The min key (use `d3.min`)
    3. The max value (use `d3.max`)
    4. Array sorted in ascending order by key (use `array.sort`, `d3.ascending`)
    5. Array sorted in descending order by value (use `array.sort`, `d3.descending`)
    6. Array containing the top 5 items sorted by value (use `array.slice`)
    7. Array with a label key as shown (use `array.map`)

```
data = [{key: "United States", value: 0.4, label: "United States (0.4)"},
 {key: "Turkey", value: 0.6, label: "Turkey (0.6)"},
 {key: "France", value: 0.2, label: "France (0.2)"},
 {key: "Italy", value: 0.1, label: "Italy (0.1)"},
 {key: "Spain", value: 0.3, label: "Spain (0.3)"},
 {key: "Algeria", value: 0.9, label: "Algeria (0.9)"},
 {key: "Mexico", value: 0.4, label: "Mexico (0.4)"},
 {key: "Brazil", value: 0.3, label: "Brazil (0.3)"},
 {key: "China", value: 0.1, label: "China (0.1)"},
 {key: "Japan", value: 0.8, label: "Japan (0.8)"}]
```

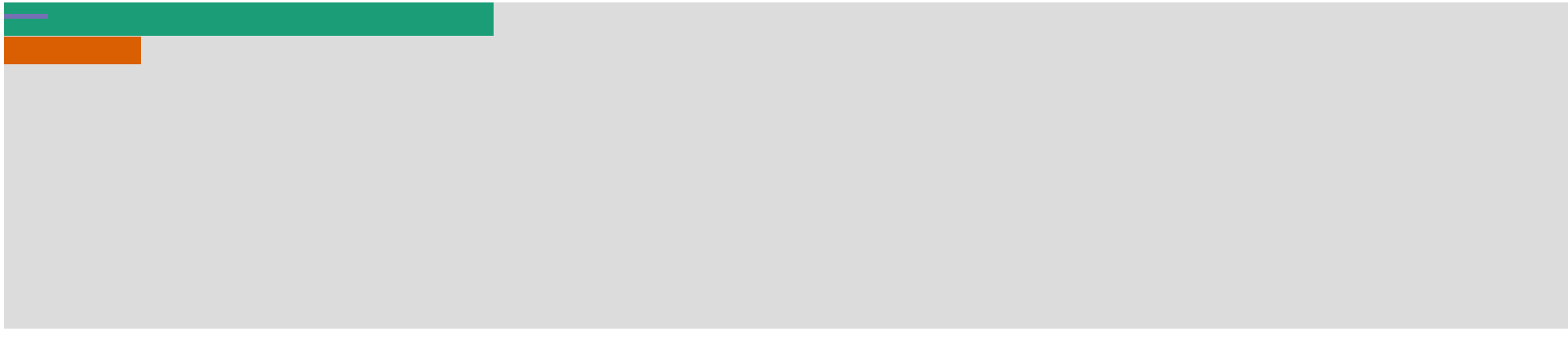
```
var array0 = [1, 4, 9, 16];
JSON.stringify(array0)  //prints "[1,4,9,16]"

var array1 = array0.map(function(x){ return x * 3; });  //array1 = [3, 12, 27, 48]
const array2 = array0.map(x => x * 3);  //same with ES6 syntax

var array3 = array0.slice(0);  //make a copy, array3 = [1, 4, 9, 16]
array3.sort();  //default sorts according to Unicode code, array3 = [1, 16, 4, 9]

var array4 = array0.slice(0);  //make a copy
array4.splice(1, 0, 5);  //insert 5 at position 1, array4 = [1, 5, 4, 9, 16]

var array5 = array0.slice(0);  //make a copy
array5.splice(1, 2);  //delete 2 elements starting at position 1, array5 = [4, 9]

var min = d3.min(array0);  //min = 1
var max = d3.max(array0);  //max = 16

var array6 = [9, 16, 4, 1];
array6.sort(d3.ascending);  //sort in place: array6 = [1, 4, 9, 16]

var array7 = [9, 16, 4, 1];
array7.sort(d3.descending);  //array7 = [16, 9, 4, 1]

var data = [{key: 'first', val: 4}, {key: 'second', val: 5}];
var min = d3.min(data, function (d) { return d.val; });  //min = 4
var max = d3.max(data, function (d) { return d.key; });  //max = 'second'

var sorted = data.sort(function(a, b) {
    return d3.descending(a.key, b.key);  //sort alphabetically Z->A
 }); //sorted = [{key: 'second', val: 5}, {key: 'first', val: 4}]
```

Sample JavaScript and D3 functions to updated data arrays

Implement the data join with general update pattern in ex2.html:

- Implement the code shown in the next page
- Adapt the code to the dataset you used in ex1.html
- Update the span background color to indicate which one is selected

dataset1 dataset2 dataset3

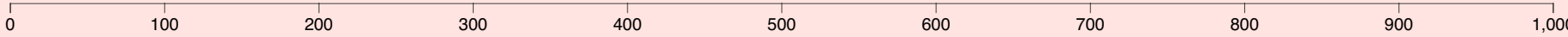


Rendering of starter code

```
<svg width="100%" height="200" style="background-color: gainsboro" id="svg_ex2"></svg>
<script>
dataset1 = [{k: "A", v: 3}, {k: "B", v: 1}, {k: "C", v: 2}];
dataset2 = [{k: "A", v: 4}, {k: "B", v: 1}, {k: "X", v: 4}];
dataset3 = [{k: "A", v: 3}, {k: "C", v: 4}, {k: "D", v: 1}, {k: "E", v: 2.5}];

update(dataset1);

d3.select("#dataset1")
  .on("click", function () {
    update(dataset1)
});

d3.select("#dataset2")
  .on("click", function () {
    update(dataset2)
});

d3.select("#dataset3")
  .on("click", function () {
    update(dataset3)
});

d3.select("#svg_ex2")
  .on("click", function() {
    update(dataset1);  //reset
});

function update(data) {
  var svg = d3.select("#svg_ex2");
  var rects = svg.selectAll("rect")
    .data(data, function (d) { return d.k; });

  rects.exit()  //EXIT SELECTION -- here we decide to exit first
    .transition()
    .delay(1000)
    .duration(1000)
    .style("opacity", 0)
    .remove();

  var enter = rects.enter()  //ENTER
    .append("rect")
    .attr("fill", function(d) { return d.c; })
    .merge(rects)  // UPDATE + ENTER
    .transition()
    .duration(3000)
    .delay(function(d, i) { return i * 1000; })
    .attr("x", 0)
    .attr("y", function(d, i) { return i * 25; })
    .attr("width", function(d) { return d.v * 100; })
    .attr("height", 20)
    .attr("fill", function(d, i) { return d3.schemeDark2[i]; });
}
</script>
```

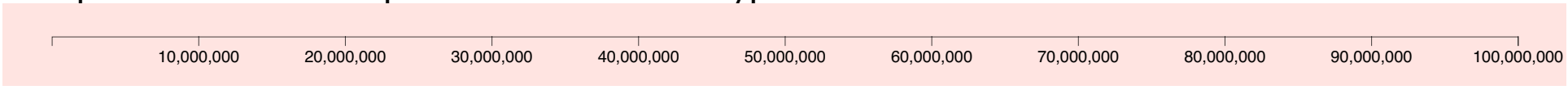
ex2.html starter code: data join with general update pattern

Complete the code to implement an axis range transition in ex3.html:




```
var x = d3.scaleLinear()
  .domain([..., ...]) //values as shown in the figure
  .range([0, 900]);

var axis1 = d3.axisBottom()
  .scale(x);

var redo = false;

var svg = d3.select("#svg");

svg.append("g")
  .attr("class", "axis0")
  .attr("transform", "translate(30,20)")
  .call(axis1);

svg.on("click", function() {
  x.domain([0, redo ? ... : ...]); //initial domain to new domain
  redo = !redo;
  d3.select(".axis0")
      .transition()
      .duration(1000)
      .call(axis1);
});
```

Complete the code to implement an axis scale type transition in ex4.html:

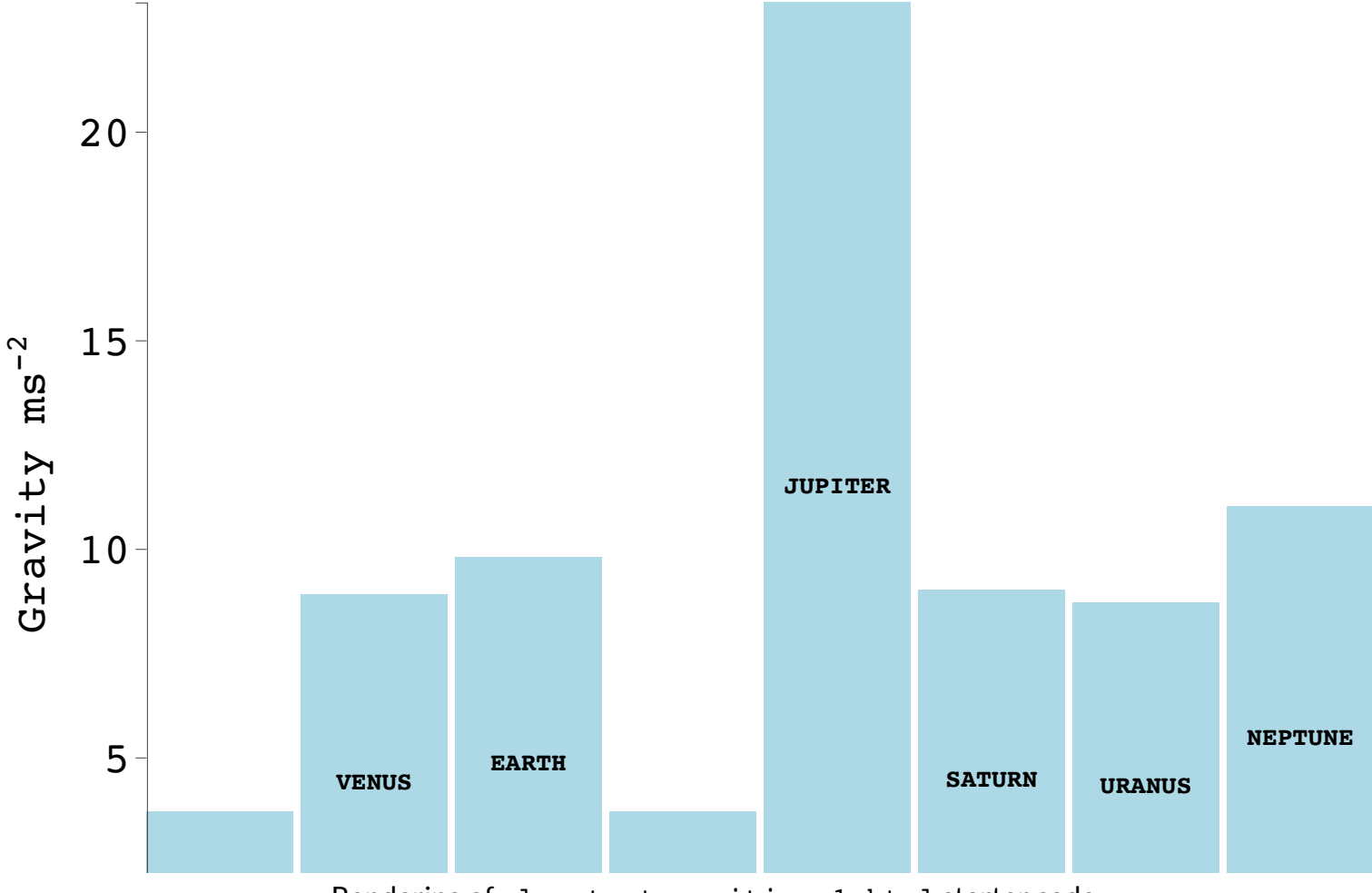



```
var x1 = d3.scaleLinear()
  .domain([..., ...])  //values as shown in the figure
  .range([0, 900]);

var x2 = d3.scaleLog()
  .domain([..., ...])  //values as shown in the figure
  .range([0, 900]);

var axis = d3.axisBottom()
  .scale(x1);

var redo = false;

var svg = d3.select("#svg");

svg.append("g")
  .attr("class", "axis0")
  .attr("transform", "translate(30,20)")
  .call(axis1);

svg.on("click", function() {
  d3.select(".axis1")
    .transition()
    .duration(1000)
    .call(redo ? axis.scale(...) : axis.scale(...)); //initial to new scale
  redo = !redo;
});
```

Implement a bar graph with transitions in ex5.html:

- Use the starter code provided in `planets_transition_1.html`
- Adapt the code to use d3 from node
- Separate the code of `ex5.html` into `ex6.html`, `ex6.css` and `ex6.js`

Order by: distance to sun, temperature, gravity ☑ Show Earth



Rendering of `planets_transition_1.html` starter code